



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Iago Zanuti Biundini

Controle de Sistemas Dinâmicos Multiagentes em Jogos Gráficos via Equilíbrio Correlacionado

Campinas

2018



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Iago Zanuti Biundini

Controle de Sistemas Dinâmicos Multiagentes em Jogos Gráficos via Equilíbrio Correlacionado

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Automação.

Orientador: Prof. Dr. Paulo Augusto Valente Ferreira

Este exemplar corresponde à versão final da tese defendida pelo aluno Iago Zanuti Biundini, e orientada pelo Prof. Dr. Paulo Augusto Valente Ferreira

Campinas

2018

Agência(s) de fomento e nº(s) de processo(s): CAPES

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

B549c Biundini, Iago Zanuti, 1993-
Controle de sistemas dinâmicos multiagentes em jogos gráficos via equilíbrio correlacionado / Iago Zanuti Biundini. – Campinas, SP : [s.n.], 2018.

Orientador: Paulo Augusto Valente Ferreira.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Sistemas de controle. 2. Sistemas multiagentes. 3. Teoria dos jogos. I. Ferreira, Paulo Augusto Valente, 1958-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Control of dynamic multiagent systems in graphical games via correlated equilibrium

Palavras-chave em inglês:

Control systems

Multi-agent systems

Game theory

Área de concentração: Automação

Titulação: Mestre em Engenharia Elétrica

Banca examinadora:

Paulo Augusto Valente Ferreira [Orientador]

Marcus Fabius Henriques de Carvalho

Pedro Luis Dias Peres

Data de defesa: 06-04-2018

Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato: Iago Zanuti Biundini RA: 163725

Data da Defesa: 06 de abril de 2018

Título da Tese: "Controle de Sistemas Dinâmicos Multiagentes em Jogos Gráficos via Equilíbrio Correlacionado".

Prof. Dr. Paulo Augusto Valente Ferreira (Presidente, FEEC/UNICAMP)

Prof. Dr. Marcio Fabius Henriques de Carvalho (PUC - Campinas)

Prof. Dr. Pedro Luis Dias Peres (FEEC/UNICAMP)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no processo de vida acadêmica do aluno.

Dedico esta tese a todos os meus amigos e estudiosos.

Agradecimentos

Gostaria de agradecer ao meu orientador, o Dr. Paulo Augusto Valente Ferreira, pela paciência, trabalho e conhecimento compartilhado nesses meses de pesquisa. Agradeço ainda a minha família e amigos pelo apoio e o imprescindível suporte da CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) durante o mestrado.

Resumo

A área de Sistemas de Controle é uma das mais antigas da Engenharia, lidando com problemas de análise e projeto de controladores (sistemas dinâmicos) com vistas a obter o comportamento desejado para o sistema dinâmico em estudo. Mais recentemente, com a emergência de sistemas dinâmicos descentralizados conectados por redes de comunicação, surgiu a necessidade de rever a estrutura de controle, que passou a contemplar tanto objetivos locais (individuais) quanto de grupos de sistemas. O principal objetivo dessa dissertação é apresentar um algoritmo de controle para sistemas dinâmicos multiagentes conectados por uma rede de comunicação utilizando Teoria dos Jogos. Este objetivo é alcançado com a noção de Equilíbrio Correlacionado em Grafos. O resultado é um algoritmo capaz de induzir um comportamento de consenso em missões de sincronização de agentes para estados pré-determinados. O algoritmo foi bem sucedido em missões de sincronização com a presença de nó líder, cujo comportamento estabelece o consenso a ser seguido pelos demais agentes do sistema.

Palavras-chaves: Sistemas de Controle; Sistemas Multiagentes; Teoria dos Jogos.

Abstract

Control systems is one of the most traditional areas in Engineering, dealing with analysis and design problems concerned to controllers (dynamic systems), aiming at obtaining a prescribed behavior for the control system under study. More recently, with the emergence of decentralized dynamic systems connected by communications networks, it became necessary to restate the structure of the control system, which, in these circumstances, needs to contemplate both local (individual) and global objectives. The main objective of this dissertation is to present a control algorithm for multi-agent dynamic systems connected by a communication network by using Game Theory. This objective is accomplished by means of the concept of Correlated Equilibrium in Graphs. The main result is an algorithm capable of inducing a consensus for agent synchronization to predetermined states. The algorithm has been well succeeded in synchronization missions in which a leader (agent) establishes the behavior (consensus) to be followed for the remaining agents.

Keywords: Control Systems; Multi-agent Systems; Game Theory.

Lista de ilustrações

Figura 1 – Exemplo de Jogo: Cruzamento.	15
Figura 2 – Exemplo de grafo.	23
Figura 3 – Exemplo de árvore geradora com raiz no nó 1.	23
Figura 4 – Jogo dos Múltiplos Cruzamentos.	27
Figura 5 – Grafo para o exemplo dos múltiplos cruzamentos.	27
Figura 6 – Ações conjuntas em 2 (V, V, V, V).	28
Figura 7 – Ações conjuntas em 2 (H, V, H, V).	29
Figura 8 – Primeira iteração do Algoritmo Iterativo - Equilíbrio de Nash em grafos	30
Figura 9 – Etapa 1 (descendente) do algoritmo TreeNash Aproximado.	31
Figura 10 – Etapa 2, Final (descendente) do algoritmo TreeNash Aproximado.	31
Figura 11 – Estrutura de um Algoritmo Genético.	37
Figura 12 – Evolução do Algoritmo Genético - Jogo dos Múltiplos Cruzamentos.	39
Figura 13 – Modelo do Algoritmo de Controle - Variação dos Estados.	41
Figura 14 – Diagrama do Algoritmo.	42
Figura 15 – Exemplo de controle dinâmico.	43
Figura 16 – Resultados do algoritmo de controle multiagentes.	45
Figura 17 – Tempo dos ensaios para 50 entradas diferentes e aleatórias.	45
Figura 18 – Representação da introdução de um nó líder.	46
Figura 19 – Resultado da introdução de um nó líder.	47
Figura 20 – Resultado para 50 entradas aleatórias para a presença do nó líder.	47
Figura 21 – Resultado da introdução de um nó líder e trajetória variante.	48
Figura 22 – Resultado da introdução de ruído e valores constantes.	48

Lista de tabelas

Tabela 1 – Matrizes do Jogo de Cruzamento.	17
Tabela 2 – Equilíbrios de Nash - Jogo do Cruzamento.	17
Tabela 3 – Solução da Otimização Quadrática - Cruzamento.	18
Tabela 4 – Matriz C do Jogo do Cruzamento.	19
Tabela 5 – Equilíbrios Correlacionados - Jogo do Cruzamento.	20
Tabela 6 – Retornos do Jogo dos Múltiplos Cruzamentos.	29
Tabela 7 – Exemplo de probabilidades para o algoritmo TreeNash Aproximado. . .	31
Tabela 8 – Linhas da restrição (3.16) do Jogo dos Múltiplos Cruzamentos.	34
Tabela 9 – Linhas da restrição (3.17) do Jogo dos Múltiplos Cruzamentos.	35
Tabela 10 – Coeficientes das restrições (3.18) relativas aos jogadores 1 e 4.	35
Tabela 11 – Solução do Jogo dos Múltiplos Cruzamentos.	36
Tabela 12 – Estrutura de representação do indivíduo - Jogo dos Múltiplos Cruzamentos.	38
Tabela 13 – Dados utilizados na simulação - Jogo dos Múltiplos Cruzamentos. . . .	38

Sumário

1	Introdução	12
2	Jogos	15
2.1	Jogos	15
2.2	Equilíbrios	16
2.2.1	Equilíbrio de Nash	16
2.2.2	Equilíbrio Correlacionado	18
3	Jogos Gráficos	21
3.1	Grafos	21
3.1.1	Matrizes de Grafos	22
3.1.2	Auto-Estrutura da Matriz Laplaciana do Grafo	24
3.1.3	O autovalor de Fiedler	26
3.2	Jogos Gráficos	26
3.2.1	Equilíbrio de Nash em Jogos Gráficos	29
3.2.2	Equilíbrio Correlacionado em Jogos Gráficos	33
3.2.3	Solução de Jogos Gráficos por Algoritmo Genético	36
4	Controle de Sistemas Dinâmicos Multiagentes	40
4.1	Algoritmos de Controle em Grafo	40
4.1.1	Estados Regulados	43
4.1.2	Consenso de Estados em Relação a um Líder	45
4.1.3	Estados Ruidosos	48
5	Conclusão	49
	Referências	50

1 Introdução

Na natureza, como nas interações sociais, cada indivíduo manifesta inclinações e movimentos que tendem a satisfazer seus interesses. O movimento de um grupo de indivíduos faz com que o grupo pareça ser uma entidade única, com suas próprias leis de movimento, psicologia e respostas a eventos externos, gerando muitas vezes resultados que o indivíduo sozinho não conseguiria. Um exemplo é a busca de aves por alimento, quando o bando atua como uma única entidade, conseguindo maior quantidade de alimento do que um único indivíduo (LEWIS *et al.*, 2013).

Considere o movimento coletivo de grandes grupos, como de aves migratórias e de outros animais. Esse movimento pode ser representado usando algumas regras simples que regem o comportamento dos indivíduos. Os movimentos individuais em um grupo são o resultado do equilíbrio de dois comportamentos opostos: desejo de ficar no grupo e de evitar colisões com outros indivíduos do grupo. Reynolds (REYNOLDS, 1987) capturou as tendências que regem os movimentos de indivíduos através de suas três regras:

- Evitar colisões: evitar colisões com os vizinhos;
- Correspondência de velocidade: combinar a velocidade e a direção do movimento com os vizinhos;
- Centralização do bando: ficar perto dos vizinhos.

As regras de Reynolds capturam muito bem o movimento coletivo de grupos de animais e também podem ser usadas como esquemas de controle para sistemas, como formações de veículos autônomos em missão coletiva, veículos espaciais, veículos aéreos não tripulados (VANT) (WANG *et al.*, 2007), sistemas multiveículos (MURRAY, 2007), robôs móveis, vigilância multiponto (CRUZ *et al.*, 2007), redes de sensores (CHEN *et al.*, 2010), equipes autônomas em rede, e assim por diante, em aplicações industriais, militares ou civis.

A necessidade do envolvimento de diversos subsistemas integrados com um objetivo global complexo tornou necessária a criação de uma nova área de estudo: Controle de Sistemas Multiagentes, uma área de estudo em que se busca a conquista de objetivos globais e individuais para um conjunto de dois ou mais sistemas ligados por uma rede de comunicação. A topologia da rede de comunicação pode limitar severamente o desempenho possível de quaisquer leis de controle usadas pelos agentes, sendo o objetivo geral criar um conjunto de protocolos distribuídos que garanta consenso (sincronização), pelo qual os

estados de todos os agentes alcancem um equilíbrio (no sentido usualmente empregado em Economia), ou seja, um estado onde os objetivos globais e individuais sejam alcançados.

Uma das formas de busca deste consenso em sistemas é através da Teoria dos Jogos. A Teoria dos Jogos tem sido utilizada na área de sistemas controle em aprendizagem de sistemas em tempo real (VAMVOUDAKIS *et al.*, 2017), formação de coalizões de sistemas (FELE *et al.*, 2017), controle distribuído (QUIJANO *et al.*, 2017), entre outras áreas. Um jogo, como será detalhado no Capítulo 2, seria uma disputa entre sistemas para que seus objetivos fossem conquistados. Um exemplo mais simples é o Jogo da Galinha. O nome do jogo tem relação com o fato de que as pessoas consideradas covardes são 'galinhas' (*chicken*, em inglês).

Os dois participantes do jogo disputam uma competição. Os jogadores posicionam seus automóveis em lados opostos de uma pista em linha reta com uma marcação na metade da pista, e devem arrancar ao mesmo tempo. Os jogadores possuem duas opções: desistir ou não desistir. Aquele que desiste, desvia do caminho; o que não desiste, segue em frente. Caso os dois oponentes não desistam, perdem tudo, incluindo possivelmente a vida. Se apenas um desiste, o que não desiste ganha, e o outro perde. E, se ambos desistem, ambos perdem o respeito dos amigos, mas ainda têm seus carros e suas vidas. Como não há, nesse jogo, a situação de ambos vencendo, é necessário definir regras para que a chance de conquista para ambos os jogadores seja a maior possível. Um outro exemplo de jogo clássico é o 'Jogo do Cruzamento', no qual dois carros disputam a passagem por um cruzamento (LUCE; RAIFFA, 2012).

Jogos Gráficos são jogos em que a topologia da rede de comunicação pode ser descrita por um grafo. Para atingir o objetivo de maximizar o retorno global do sistema, no caso a soma dos retornos individuais de cada jogador, é necessária uma definição apropriada do grafo, além de conceitos de equilíbrio que caracterizam o resultado do jogo.

A proposta dessa dissertação é desenvolver algoritmos para obtenção de equilíbrios em Jogos Gráficos (Nash e Correlacionado) utilizando algoritmos clássicos e um algoritmo genético. Propõe-se ainda um algoritmo para controle dinâmico multiagentes utilizando os conceitos de Equilíbrio Correlacionado em Grafos.

Essa dissertação está organizada em cinco capítulos.

No Capítulo 2 são apresentadas noções de jogos entre dois jogadores, os chamados Jogos Bimatrizes. São apresentadas os algoritmos para obtenção dos equilíbrios de Nash e Correlacionado, aplicado ao Jogo do Cruzamento, além de uma comparação entre os equilíbrios de Nash e Correlacionado.

No Capítulo 3 são apresentadas noções de Jogos Gráficos. É destacada a importância da topologia escolhida, e introduzidos alguns conceitos sobre análise algébrica e representação de grafos. É realizada a implementação de algoritmos para obtenção dos

equilíbrios de Nash e Correlacionado. A partir de um exemplo, o Jogo dos Multicruzamentos, é realizada uma comparação entre os equilíbrios de Nash e Correlacionado, obtidos tanto por algoritmos clássicos quanto por um algoritmo genético proposto.

No capítulo 4 são apresentadas noções sobre representação de sistemas em grafos, além de uma proposta de controle de sistemas dinâmicos multiagentes baseada no Equilíbrio Correlacionado. Resultados computacionais demonstram a eficácia e eficiência da estrutura de controle proposta.

No Capítulo 5 apresentamos as conclusões gerais do trabalho e temas para possíveis trabalhos futuros.

2 Jogos

Neste capítulo serão apresentadas noções de jogos entre dois jogadores, os conceitos de equilíbrio de Nash e equilíbrio Correlacionado e uma comparação entre eles. São também apresentados os algoritmos para obtenção destes equilíbrios aplicados a um exemplo clássico da literatura.

2.1 Jogos

Um Jogo é uma situação estratégica em que os jogadores escolhem ações visando maximizar seus retornos. O retorno de cada jogador depende de sua ação individual e das ações dos demais jogadores. Um exemplo de jogo, ilustrado na Figura 1, surge em um cruzamento viário. Cada jogador, no caso um carro na via, quer conquistar seu objetivo (atravessar a via), porém os carros não podem fazer isso ao mesmo tempo. Esse exemplo irá ilustrar todo o nosso estudo sobre jogos entre dois jogadores, os chamados Jogos Bimatrizes.

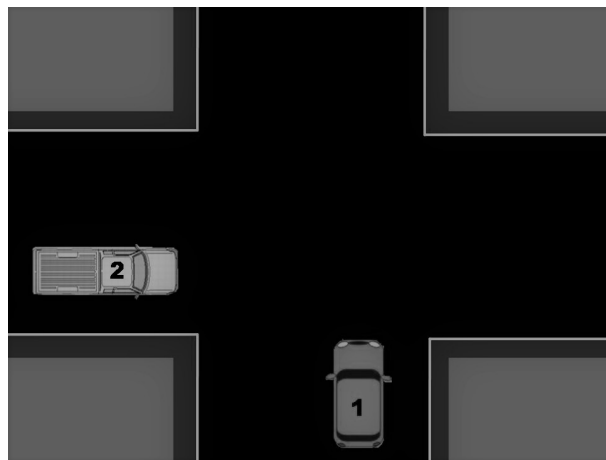


Figura 1 – Exemplo de Jogo: Cruzamento.

Em Teoria dos Jogos (PETERS, 2015), distinguimos dois tipos de estratégias: puras, como avançar ou parar no cruzamento, ou mistas, ao atribuir probabilidades para a adoção de uma ação pura, como: escolha avançar com probabilidade 0,7, e parar, com probabilidade 0,3. Em outras palavras, uma estratégia mista do jogador i é uma distribuição de probabilidades sobre estratégias puras. No caso de estratégias binárias, uma estratégia pura é escolher 0 ou 1. Numa estratégia mista, escolhe-se 0 com probabilidade $p \geq 0$ e 1, com probabilidade $1 - p$.

Denotando como A_i o conjunto de ações puras do jogador i , o vetor $\vec{a} = (a_1, a_2, \dots, a_n)$ representa a ação conjunta de n jogadores, também chamada de perfil estratégico, onde

$a_i \in A_i$. No caso binário, temos $A_i = \{0, 1\}$. Para três jogadores, uma possível ação conjunta é $\vec{a} = (0, 1, 0)$: o jogador 1 escolhe 0, o jogador 2 escolhe 1 e o jogador 3 escolhe 0.

Uma importante noção em jogos é o retorno de cada jogador. Como o objetivo do jogo é alcançar o maior retorno possível para o jogador, definimos a matriz $M_i(\vec{a})$ que representa os retornos do jogador i para cada ação conjunta \vec{a} . O retorno do jogador i não depende somente da ação do jogador i . Aqui, o termo retorno é sinônimo de utilidade, e o objetivo de cada jogador é maximizar sua utilidade dadas as ações dos demais jogadores.

A melhor resposta a um conjunto de perfis estratégicos para o jogador i é uma ação \vec{a} tal que $M_i(\vec{a}) \geq M_i(\vec{a}')$, onde \vec{a}' é qualquer outro perfil estratégico possível.

2.2 Equilíbrios

O objetivo geral em jogos é levar os jogadores a uma condição de equilíbrio. Os equilíbrios em jogos mais conhecidos são os equilíbrios de Nash e Correlacionado.

2.2.1 Equilíbrio de Nash

O Equilíbrio de Nash (NASH, 1951) descreve a situação em que, em um jogo envolvendo dois ou mais jogadores, nenhum jogador tem incentivo (aumento do seu retorno próprio) para mudar sua ação unilateralmente, caso todos os outros jogadores mantenham suas ações.

Um jogo de dois jogadores na chamada forma normal pode ser representado por duas matrizes $m \times n$ (A, B), onde m (número de ações puras do jogador 1) é o número de linhas e n (número de ações puras do jogador 2) o número de colunas de A e B . A interpretação de um jogo bimatriz (A, B) é que, se o jogador 1 (o jogador linha) escolher a linha i e o jogador 2 (o jogador coluna) escolher a coluna j , então o jogador 1 recebe a recompensa a_{ij} e o jogador 2 recebe b_{ij} , onde a_{ij} e b_{ij} são os coeficientes correspondentes de A e B (PETERS, 2015). Para esse jogo, o equilíbrio de Nash pode ser calculado por meio de Otimização Quadrática descrita pela formulação:

$$\begin{aligned} & \underset{p \in \Delta^m; q \in \Delta^n; a, b \in \mathbb{R}}{\text{Maximizar}} && f(p, q, a, b) := pAq + pBq - a - b \\ & \text{sujeito a} && e^i Aq \leq a, \quad i = 1, \dots, m. \\ & && pAe^j \leq b, \quad j = 1, \dots, n \end{aligned} \tag{2.1}$$

onde $\Delta^k = \{p = (p_1, \dots, p_k) : p_i \geq 0, \forall i; \sum_{i=1}^k p_i = 1\}$, e^i é uma estratégia pura do jogador i , ou seja, $p_i = 1$ e e^j é uma estratégia pura do jogador j , ou seja, $q_j = 1$

As soluções ótimas de (2.1) p^* e q^* caracterizam o equilíbrio de Nash para os jogadores 1 e 2, respectivamente. Verifica-se que $a^* = p^* A q^*$ e $b^* = p^* B q^*$ são os retornos ótimos para os jogadores 1 e 2, respectivamente.

No caso do Jogo do Cruzamento (Figura 1), inicialmente temos que definir as matrizes (A, B) . Definindo as estratégias puras como passar e parar, temos as seguintes situações:

- Ambos os carros passam: ocorrerá colisão, a situação de pior caso. Os retornos dos jogadores são -10 .
- Carro 1 passa e o Carro 2 para: o objetivo do jogador 1 é conquistado com retorno 5. O jogador 2 não terá prejuízo, pois os recursos da via foram usados, e seu retorno é 0.
- Carro 2 passa e o varro 1 para: caso análogo ao item anterior; retorno 5 para o jogador 2 e 0 para o jogador 1.
- Ambos os carros param. Nesse caso há um desperdício dos recursos da via, que ninguém está passando por ela. Neste caso, ambos recebem retorno -1 , melhor do que na colisão.

Dessa forma, definimos as seguintes matrizes A e B , como na Tabela 1.

Tabela 1 – Matrizes do Jogo de Cruzamento.

Jogador 1 (A)	Carro 2 Passar	Carro 2 Parar	Jogador 2 (B)	Carro 2 Passar	Carro 2 Parar
Carro 1 Passar	-10	5	Carro 1 Passar	-10	0
Carro 1 Parar	0	-1	Carro 1 Parar	5	-1

Os equilíbrios de Nash deste jogo em ações puras estão indicados na Tabela 2. $p(\text{Ação})$ representa a probabilidade da ação.

Tabela 2 – Equilíbrios de Nash - Jogo do Cruzamento.

	$p(\text{Passar})$	$p(\text{Parar})$		$p(\text{Passar})$	$p(\text{Parar})$
Jog.1	1	0	Jog.1	0	1
Jog.2	0	1	Jog.2	1	0

O jogador 1 passa (primeira matriz) ou o jogador 2 passa (segunda matriz). Caso o jogador 2 mude sua estratégia de parar para passar, seu retorno mudará de 0 para -10 , já que o jogador 1 manteve sua estratégia de passar. Obtém-se um equilíbrio de Nash. O outro equilíbrio é alcançado de maneira análoga.

Tabela 3 – Solução da Otimização Quadrática - Cruzamento.

p^*	3/8	5/8
q^*	3/8	5/8

Resolvendo o problema de otimização quadrática (2.1), obtemos os resultados da Tabela 3. Os carros devem passar com probabilidade 3/8 e parar com probabilidade 5/8. O retorno do jogador 1 ($a^* = p^* A q^*$) é $-5/8$, o mesmo do jogador 2 ($b^* = q^* B p^*$), neste exemplo. Isso significa que se ambos os jogadores adotarem estas probabilidades, ou seja, executarem suas ações de acordo com a probabilidades fornecidas pelos equilíbrios, o retorno esperado será de $-5/8$ para ambos.

2.2.2 Equilíbrio Correlacionado

O Equilíbrio Correlacionado (AUMANN, 1987) introduz um dispositivo de comunicação externo, que irá atribuir probabilidades para cada ação conjunta possível. Se todos os jogadores adotarem as probabilidades sugeridas, todos sairão ganhando, aumentando seus retornos individuais relativamente aos obtidos com o equilíbrio de Nash.

No exemplo do cruzamento, o equilíbrio correlacionado pode ser interpretado como a introdução de um semáforo no cruzamento, o qual indicará as ações Passar e Parar para os jogadores. Agora não serão mais observadas apenas as ações dos jogadores, e sim as ações conjuntas dos jogadores, isto é, não será atribuída uma probabilidade do jogador i avançar, e sim a probabilidade do jogador i e demais jogadores realizarem ações conjuntas (por exemplo, os jogadores i e j avançarem). Essa forma de computar as probabilidades correlaciona as ações dos jogadores.

Considerando as mesmas matrizes (A, B) da Tabela 1, o equilíbrio correlacionado pode ser visto como uma matriz $m \times n$ $P = (p_{ij})$, no qual $\sum_{i=1}^m \sum_{j=1}^n p_{ij} = 1$ e $p_{ij} \geq 0$, para todo $i = 1, \dots, m$ e $j = 1, \dots, n$. O par (i, j) é escolhido com probabilidade p_{ij} , em que o jogador 1 escolhe a ação i e o jogador 2 a ação j .

Para que seja vantajoso ao jogador i adotar a ação recomendada, ou seja, não mudar unilateralmente sua ação, é necessário que:

$$\sum_{j=1}^n (a_{ij} - a_{kj}) p_{ij} \geq 0 \quad (2.2)$$

para todo $i, k = 1, \dots, m$. O retorno esperado com a ação i é maior ou igual ao retorno esperado com a ação k .

Analogamente, para o jogador 2:

$$\sum_{i=1}^m (b_{ij} - b_{il}) p_{ij} \geq 0 \quad (2.3)$$

Tabela 4 – Matriz C do Jogo do Cruzamento.

C	-10	0	-10	0
	6	0	0	10
	0	10	6	0
	0	-6	0	-6

para todo $j, l = 1, \dots, n$.

Uma solução que satisfaz as desigualdades (2.2) e (2.3), além de $\sum_{i=1}^m \sum_{j=1}^n p_{ij} = 1$ e $p_{ij} \geq 0$ para todo $i = 1, \dots, m$ e $j = 1, \dots, n$, é um equilíbrio correlacionado do jogo.

Uma forma de calcular essas probabilidades é através de Programação Linear. Dadas as matrizes $(A, B), m \times n$, a estas matrizes é associada outra matriz, $C, mn \times (m(m-1) + n(n-1))$. Para cada par (i, j) de uma linha e uma coluna em (A, B) , temos uma linha em C e para cada par (h, k) de duas linhas diferentes em (A, B) ou duas colunas diferentes em (A, B) , temos uma coluna em C da seguinte forma:

$$C_{(i,j)(h,k)} = \begin{cases} a_{ij} - a_{kj}, & \text{se } i = h \in \{1, \dots, m\} \text{ e } k \in \{1, \dots, m\} \\ b_{ij} - b_{ik}, & \text{se } j = h \in \{1, \dots, n\} \text{ e } k \in \{1, \dots, n\} \\ 0, & \text{caso contrário.} \end{cases} \quad (2.4)$$

Desta forma, $C_{h,k}$ é uma coluna de C . Se h e k são linhas de (A, B) temos:

$$p^T C_{h,k} = \sum_{j=1}^n p_{hj} (a_{hj} - a_{kj}) \quad (2.5)$$

e se h e k forem colunas de (A, B) :

$$p^T C_{h,k} = \sum_{i=1}^m p_{ih} (b_{ih} - b_{ik}) \quad (2.6)$$

que representam as desigualdades (2.2) e (2.3). Dessa forma, P , que representa as probabilidades de cada ação conjunta ser adotada, é um equilíbrio correlacionado se $P^T C \geq 0$. Uma função-objetivo usual para o problema de Programação Linear é definida pela soma de todos os retornos de todos os jogadores.

No Jogo do Cruzamento com as matrizes (A, B) dadas pela Tabela 1, obtemos a matriz da Tabela 4. A matriz P representa as probabilidades de ambos seguirem, um seguir e o outro passar, vice e versa, e ambos ficarem parados. Na Tabela 5 apresentamos os equilíbrios correlacionados do Jogo do Cruzamento.

Os dois primeiros equilíbrios têm retorno global (soma de todos os retornos individuais) igual a $5 = (0 \times -10) + (1 \times 5) + (0 \times 0) + (0 \times -1)$; o terceiro equilíbrio tem retorno global 5, com 2,5 para cada jogador $((0 \times -10) + (0,5 \times 5) + (0,5 \times 0) + (0 \times -1))$. Qualquer solução que atenda (2.2) e (2.3) é um equilíbrio correlacionado. Os retornos dos

Tabela 5 – Equilíbrios Correlacionados - Jogo do Cruzamento.

	Jog.2 Passar	Jog.2 Parar		Jog.2 Passar	Jog.2 Parar		Jog.2 Passar	Jog.2 Parar
Jog.1 Passar	0	1	Jog.1 Passar	0	0	Jog.1 Passar	0	0,5
Jog.1 Parar	0	0	Jog.1 Parar	1	0	Jog.1 Parar	0,5	0

equilíbrios correlacionados de maiores retornos são sempre maiores ou iguais aos retornos do equilíbrio de Nash, pois o equilíbrio de Nash é um equilíbrio correlacionado com P definida pelos produtos das probabilidades (PETERS, 2015).

3 Jogos Gráficos

Os jogos discutidos no Capítulo 2 são conhecidos como Jogos Bimatrizes, pois têm seus retornos representados por duas matrizes, (A, B) , que definem toda a estrutura do jogo. A comunicação é feita entre os dois jogadores de forma bidirecional. Cada jogador conhece as ações e retornos do outro jogador. Quando o número de jogadores é maior do que 2, as informações disponíveis para cada jogador podem estar restritas a uma certa 'vizinhança' de jogadores. Dessa forma é adicionado mais um elemento ao estudo de jogos: A topologia da rede de comunicação. Neste capítulo apresentamos noções sobre a representação de redes por grafos, além de extensões dos conceitos de jogos e equilíbrios. Serão apresentados algoritmos que obtêm os equilíbrios tanto por métodos clássicos quanto por Programação Evolutiva.

3.1 Grafos

Um grafo representa o fluxo de informações permitido entre os jogadores, aqui também chamados de agentes. Um grafo é um par $\mathcal{G} = (V, E)$, com $V = \{v_1, \dots, v_N\}$ o conjunto de N nós ou vértices e E o conjunto de arestas ou arcos. Elementos de E são denotados por (v_i, v_j) , que é um arco do nó v_i para o nó v_j e representado por uma seta com origem em v_i apontando para v_j .

O grau interno (*in-degree*) de v_i é o número de arestas que terminam em v_i , ou seja, o número de arestas que apontam para v_i . O grau externo (*out-degree*) de um nó v_i é o número de arestas que têm v_i como origem, ou seja, o número de arestas que saem de v_i . Se o *in-degree* for igual ao *out-degree* para todos os nós $v_i \in V$, o grafo é dito balanceado. Se $(v_i, v_j) \in E \Rightarrow (v_j, v_i) \in E, \forall i, j$, o grafo é bidirecional ou um dígrafo.

A vizinhança do nó i , denotada por N_i , é o conjunto de nós constituído por todos os nós adjacentes a i . A vizinhança pode incluir ou não o nó i , sendo uma vizinhança aberta quando não inclui, e fechada, caso contrário. Nesta dissertação, a vizinhança é considerada fechada, em geral, pois as ações conjuntas na região do jogador i dependem da ação deste mesmo jogador.

Associada a cada aresta (v_i, v_j) pode existir um peso a_{ij} (a ordem dos índices ij mostra que o peso é da aresta v_i para v_j) que representa o custo de ir do vértice v_i para v_j . O grafo é não direcionado se $a_{ij} = a_{ji}, \forall i, j$, isto é, um grafo bidirecional em que os pesos de (v_i, v_j) e (v_j, v_i) são iguais.

Um caminho é uma sequência finita de nós conectados por arestas. Desta forma, o caminho de um nó i para o nó j é o conjunto de nós entre os nós i e j , representando

uma conexão entre os nós levando em conta se a aresta é direcionada ou bidirecional. Um caminho mínimo é um caminho no qual a somatória dos custos das arestas é mínimo. A distância entre dois nós em um grafo é o comprimento do caminho mínimo, se houver. Caso contrário, a distância entre os nós é infinita.

Um grafo é conexo se há caminhos conectando cada par de nós do grafo. Um grafo é fortemente conectado se há caminhos direcionados opostos contendo cada par de nós.

Uma árvore é um dígrafo conectado no qual cada nó, exceto um, chamado raiz, possui *in-degree* igual a 1. Uma árvore geradora de um dígrafo é uma árvore direcionada formada por arcos que conectam todos os nós do grafo. Um grafo é uma árvore geradora se um subconjunto de arestas forma uma árvore direcionada. Isso equivale a dizer que todos os nós no grafo são acessíveis a partir de um único nó (raiz), seguindo as setas dos arcos. Um grafo fortemente conectado contém pelo menos uma árvore geradora. Em um grafo fortemente conectado, cada nó pode ser definido como nó raiz.

3.1.1 Matrizes de Grafos

A estrutura do grafo e suas propriedades podem ser estudadas examinando-se certas matrizes associadas ao grafo, o que caracteriza a Teoria Algébrica dos Grafos.

Dados os pesos a_{ij} , o grafo pode ser representado por uma matriz de adjacência, ou conectividade, $A = (a_{ij})$, com $a_{ij} > 0$ se $(v_i, v_j) \in E$, e $a_{ij} = 0$, caso contrário.

Defina o *in-degree* ponderado do nó v_i pela soma da i -ésima linha de A :

$$d_i = \sum_{j=1}^N a_{ij} \quad (3.1)$$

Defina o *out-degree* ponderado do nó v_i pela soma da i -ésima coluna de A :

$$d_i^o = \sum_{j=1}^N a_{ji} \quad (3.2)$$

O *in-degree* e *out-degree* são propriedades locais do grafo. Duas importantes propriedades globais são o diâmetro ($\text{Diam } G$), que é a maior distância entre dois nós do grafo, e o volume interno do grafo ($\text{Vol } G$), que é a soma dos *in-degrees* dos nós do grafo:

$$\text{Vol } G = \sum_{i=1}^N d_i \quad (3.3)$$

A matriz de adjacência A para um grafo não direcionado é simétrica, $A = A^T$. Um grafo é balanceado em pesos se o *in-degree* ponderado e *out-degree* ponderado são iguais para todo i . Um grafo não direcionado é balanceado em pesos, dado que $A = A^T$, pois

a soma da i -ésima linha é igual a da i -ésima coluna de A . Por simplicidade, às vezes nos referimos ao nó v_i simplesmente como nó i , e a *in-degree*, *out-degree* sem o qualificador 'em pesos'.

Definida a matriz diagonal *in-degree* $D = \text{diag}\{d_i\}$, a matriz Laplaciana do grafo é $L = D - A$. Note que a soma de todas as linhas de L é igual a zero. Muitas propriedades de um grafo podem ser estudadas a partir de sua matriz Laplaciana. A matriz Laplaciana é de extrema importância no estudo de sistemas dinâmicos multiagentes em grafos.

Como exemplo, considere o grafo da Figura 2 no qual todos os pesos dos arcos são iguais a 1. O grafo é fortemente conectado porque existe um caminho entre qualquer dois pares de nós. Na Figura 3, em negrito, selecionamos uma árvore geradora. Existem outras árvores geradoras no grafo, onde cada nó pode ser raiz, já que o grafo é fortemente conectado.

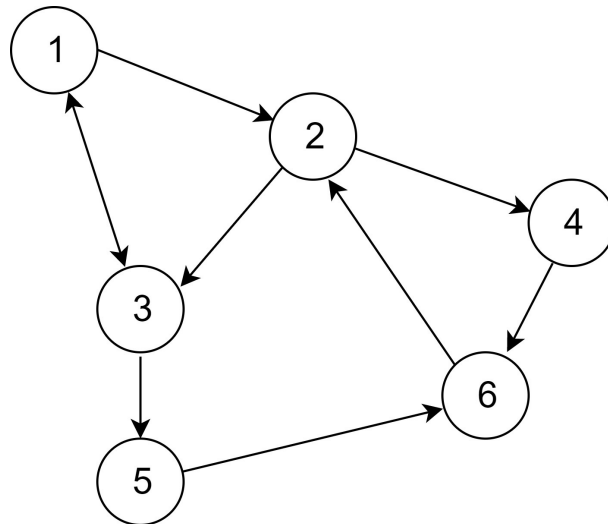


Figura 2 – Exemplo de grafo.

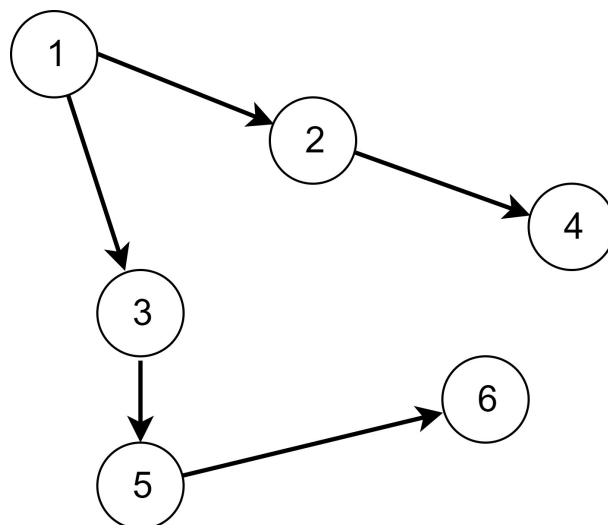


Figura 3 – Exemplo de árvore geradora com raiz no nó 1.

A matriz de adjacência é dada por (3.4). As matrizes diagonal *in-degree* e Laplaciana são representadas por (3.5) e (3.6), respectivamente. Note que a soma de cada linha de L é igual a zero.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (3.4)$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad (3.5)$$

$$L = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 & -1 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix} \quad (3.6)$$

3.1.2 Auto-Estrutura da Matriz Laplaciana do Grafo

A auto-estrutura da matriz Laplaciana desempenha um papel fundamental na análise do grafo. Defina a forma de Jordan normal da matriz Laplaciana como:

$$L = MJM^{-1} \quad (3.7)$$

onde a matriz de Jordan J e a matriz de transformação M são dadas por:

$$J = \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_k \end{bmatrix}, \quad M = [v_1, v_2, \dots, v_N], \quad (3.8)$$

e os autovalores λ_i e os autovetores a direita v_i de L satisfazem:

$$(\lambda_i I - L)v_i = 0, \quad (3.9)$$

sendo I a matriz identidade.

Os blocos de Jordan J_k de L possuem a forma geral:

$$J_k = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix}, \quad k = 1, 2, \dots, k, k \leq N \quad (3.10)$$

O número de blocos de Jordan associado ao mesmo autovalor λ_i é conhecido como a multiplicidade geométrica do autovalor λ_i . A soma dos tamanhos de todos os blocos de Jordan associados aos autovalores λ_i é chamada de multiplicidade algébrica de λ_i .

Para facilidade de notação e discussão, assumimos aqui que a forma do Jordão é simples, ou seja, é diagonal com todos os blocos do Jordan de tamanho 1. Isto é garantido se todos os autovalores de L forem distintos. Uma matriz simétrica pode não possuir autovalores distintos, mas tem uma forma simples de Jordan.

A inversa da matriz de transformação M é dada por:

$$M^{-1} = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_N^T \end{bmatrix} \quad (3.11)$$

onde os autovetores à esquerda w_i satisfazem:

$$w_i^T (\lambda_i I - L) = 0 \quad (3.12)$$

e são normalizados, ou seja, $w_i^T v_i = 1$.

Assumimos que os autovalores são ordenados, ou seja, $|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_N|$. Qualquer grafo não direcionado tem $L = L^T$. Neste caso, todos os autovalores são reais e podem ser ordenados da forma $|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_N|$. Como L tem todas as somas de linhas iguais a zero, então:

$$L \underline{1} c = 0 \quad (3.13)$$

onde $\underline{1} = [1, \dots, 1]^T \in R^N$ é um vetor de uns e c uma constante. Portanto, $\lambda_1 = 0$ é um autovalor com um autovetor à direita $\underline{1}c$, ou seja, $\underline{1}c \in \mathcal{N}(L)$ (espaço nulo de L). Se a dimensão do espaço nulo de L é igual a um, isto é, o rank de L é $N - 1$, então $\lambda_1 = 0$ não se repete e $\underline{1}c$ é o único vetor em $\mathcal{N}(L)$. Dessa forma temos o seguinte teorema.

Teorema 3.1 L possui rank $N - 1$, isto é, $\lambda_1 = 0$ não se repete, se e somente se o grafo G possui uma árvore geradora.

Como exemplo, considere o grafo da Figura 2, com todos os pesos iguais a 1. Os seus autovalores são: $\{0, 0.78, 1, 2.24 + 1.03i, 2.23 - 1.03i, 2.72\}$. Caso o mesmo grafo fosse não direcionado, teríamos os seguintes autovalores: $\{0, 1.38, 1.69, 3.62, 4, 5.30\}$. A matriz Laplaciana tem somas de linhas iguais a zero. Desta forma, todos os grafos tem o primeiro autovalor $\lambda_1 = 0$. Todos os grafos não direcionados têm uma matriz Laplaciana simétrica L e, portanto, seus autovalores são reais.

3.1.3 O autovalor de Fiedler

O segundo autovalor λ_2 da matriz Laplaciana L é importante para determinar a intensidade da conexão de sistemas dinâmicos em grafos. As topologias de grafos que têm um grande valor de λ_2 são melhores para se alcançar a convergência para um consenso, por exemplo. O segundo autovalor de L é conhecido como o autovalor de Fiedler e possui conexões importantes e sutis com a topologia do grafo.

Para grafos não direcionados, existe um limite útil para o autovalor Fiedler dado por:

$$\lambda_2 \leq \frac{N}{N-1} d_{min} \quad (3.14)$$

onde d_{min} é o menor *in-degree*. Para grafos conectados não direcionados,

$$\lambda_2 \geq \frac{1}{DiamG \times VolG} \quad (3.15)$$

3.2 Jogos Gráficos

A partir das definições básicas do Capítulo 2, podemos definir Jogo Gráfico como um par $(\mathcal{G}, \mathcal{M})$ onde \mathcal{G} é um grafo com vértices $\{1, \dots, n\}$ e \mathcal{M} é um conjunto de n matrizes M_i de retornos, uma para cada jogador. Para cada ação conjunta, \vec{a} , a matriz M_i especifica os retornos $M_i(\vec{a})$ do jogador i , que depende somente das ações dos jogadores na vizinhança N_i . A notação $\vec{a}_i^k = s$ representa a ação conjunta da vizinhança de i na qual o jogador $k \in N_i$ usa a estratégia pura s .

A melhor resposta para um conjunto de perfis estratégicos é uma ação \vec{a}_i tal que $M_i(\vec{a}_i) \geq M_i(\vec{a}_i')$, onde \vec{a}_i' é qualquer outro perfil estratégico da vizinhança N_i , ou seja, \vec{a}_i é o perfil estratégico que tem maior retorno na vizinhança para o jogador i .

Um jogo gráfico que generaliza o Jogo do Cruzamento do Capítulo 2 com um conjunto de vias com cruzamentos é ilustrado na Figura 4.

Para a modelagem, cada nó do grafo representará um jogador que controla o semáforo do cruzamento. O grafo possui cinco nós. As ações puras permitidas serão duas:

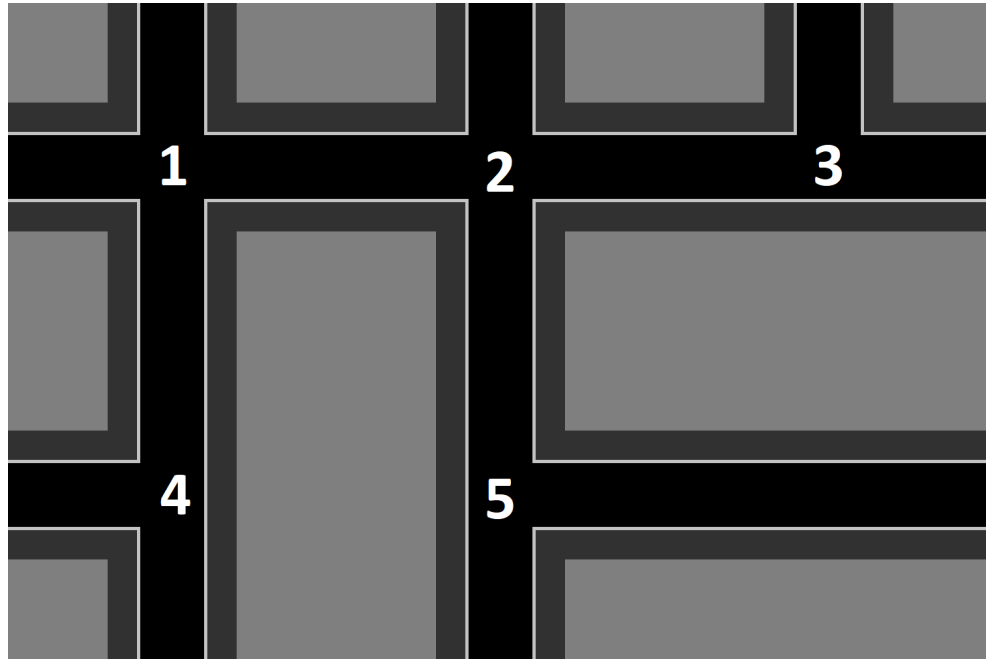


Figura 4 – Jogo dos Múltiplos Cruzamentos.

'Seguir Verticalmente' (Ação 0 ou V), onde o fluxo de veículos será na vertical dos cruzamentos, ou 'Seguir Horizontalmente' (Ação 1 ou H), onde o fluxo será na horizontal. Dessa forma, caso uma ação seja escolhida, a outra ação será bloqueada (se no nó i a ação for V , o sentido horizontal será bloqueado). Modelos mais complexos podem ser definidos, mas para uma abordagem inicial a escolha de ações binárias é de mais fácil aplicação e manipulação.

O jogo da Figura 4 tem a rede de comunicação representada pelo grafo da Figura 5.

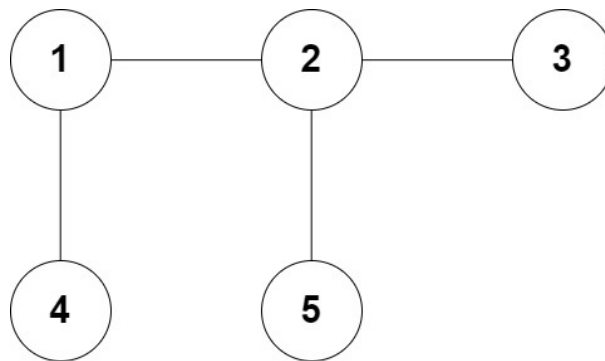


Figura 5 – Grafo para o exemplo dos múltiplos cruzamentos.

As matrizes de retorno derivam das seguintes hipóteses:

- Ações na vizinhança do nó i que levam a fluxo no nó i no mesmo sentido da ação do nó i adicionam $+1$ ao retorno de i .

- Ações na vizinhança do nó i que levem a fluxos no nó i no seu sentido contrário à ação do nó i adicionam -1 ao retorno de i .
- Ações na vizinhança do nó i que não levem a fluxo no nó i somam 0 ao retorno de i .

Desta forma, se uma ação ajudar a melhorar o fluxo no nó i , haverá um retorno positivo no nó i . Caso haja acúmulo de veículos, a ação terá um retorno negativo; ações desvinculadas levarão a retornos nulos.

Considere, por exemplo, o jogador (nó) 2 na Figura 6, que tem como vizinhos os jogadores $\{1, 2, 3, 5\}$. Toda ação conjunta nesta vizinhança envolverá 4 números. O jogador 1 tem como ação 'Seguir Verticalmente', porém essa ação não leva fluxo para o jogador 2, adicionando 0 ao retorno de 2. O mesmo ocorre com o jogador 3 (setas azuis). Já para o jogador 5, que também tem a ação 'Seguir Verticalmente', sua ação envia fluxo na mesma direção do jogador 2, desta forma adicionando $+1$ ao retorno do jogador 2 (seta verde). Assim, o retorno do jogador 2 para a ação conjunta (V, V, V, V) será $+1$ ($0 + 0 + 1$).

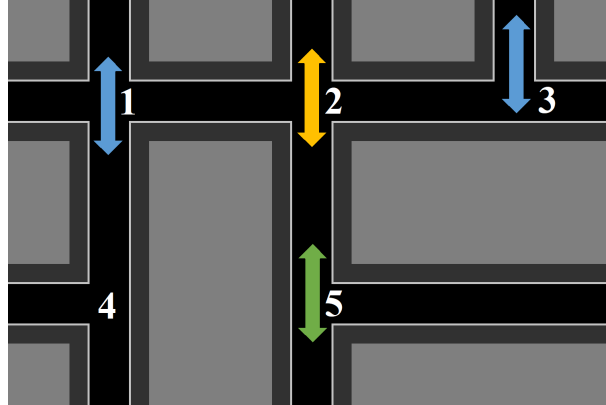


Figura 6 – Ações conjuntas em 2 (V, V, V, V) .

Se a ação conjunta for (H, V, H, V) , como mostrado na Figura 7, o jogador 1 possui a ação 'Seguir Horizontalmente' que envia o fluxo na direção contrária à do jogador 2, acumulando veículos no cruzamento de 2, desta forma adicionando -1 ao retorno do jogador 2. O mesmo ocorre com a ação do jogador 3 (setas vermelhas). O jogador 5 continua enviando fluxo na mesma direção do jogador 2, adicionando $+1$ ao retorno de 2 (Seta Verde). Desta forma, para a ação conjunta (H, V, H, V) , o retorno do jogador 2 será -1 ($-1 - 1 + 1$). Na Tabela 6 apresentamos os valores de todos os retornos para cada uma das possíveis ações conjuntas do exemplo.

Os retornos dos jogadores variarão entre -2 e $+2$. As próximas seções discutem como obter os equilíbrios de Nash e Correlacionado para os retornos da Tabela 6.

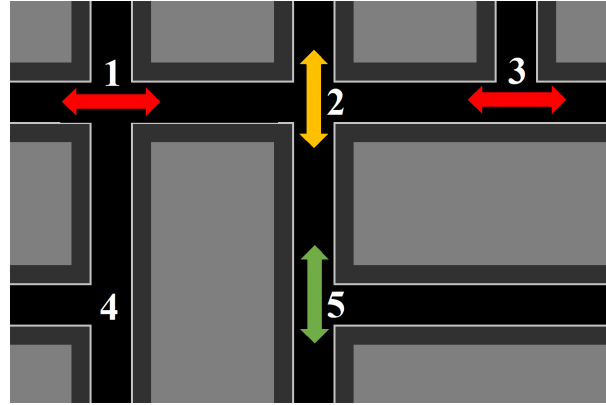
Figura 7 – Ações conjuntas em 2 (H, V, H, V).

Tabela 6 – Retornos do Jogo dos Múltiplos Cruzamentos.

	(V,V,V)	(V,V,H)	(V,H,V)	(V,H,H)
Jogador 1 {1,2,4}	1	0	0	-1
	(H,V,V)	(H,V,H)	(H,H,V)	(H,H,H)
	-1	0	0	1
Jogador 2 {1,2,3,5}	(V,V,V,V)	(V,V,V,H)	(V,V,H,V)	(V,V,H,H)
	1	0	0	-1
	(V,H,V,V)	(V,H,V,H)	{V,H,H,0}	(V,H,H,H)
	-1	0	0	1
	(H,V,V,V)	(H,V,V,H)	(H,V,H,V)	(H,V,H,H)
	0	-1	-1	-2
	(H,H,V,V)	(H,H,V,H)	(H,H,H,V)	(H,H,H,H)
	0	1	1	2
Jogador 3 {2,3}	(V,V)	(V,H)	(H,V)	(H,H)
	0	0	-1	1
Jogador 4 {1,4}	(V,V)	(V,H)	(H,V)	(H,H)
	1	-1	0	0
Jogador 5 {2,5}	(V,V)	(V,H)	(H,V)	(H,H)
	1	-1	0	0

3.2.1 Equilíbrio de Nash em Jogos Gráficos

O Equilíbrio de Nash é definido como um conjunto de ações dos jogadores a partir do qual não é vantajoso para nenhum jogador mudar sua ação unilateralmente se os demais jogadores mantiverem as ações prescritas no equilíbrio. Dentre os algoritmos usados para obter equilíbrios de Nash, destacamos dois: Algoritmo Iterativo e TreeNash. Os algoritmos discutidos a seguir foram implementados em Matlab V2009.

O primeiro algoritmo (JACKSON, 2010) assume que o jogo é binário, ou seja, o conjunto de ações puras de cada jogador é $A_i = \{0, 1\}$. O algoritmo, apresentado no Pseudocódigo 1, consiste em inicializar todas as ações em 1, e a partir disso, alterar a ação de um jogador para 0, verificando se essa modificação traz melhor retorno para o jogador. Se sim, atualiza-se o vetor de ações com o jogador usando a ação 0, e passa-se

para o próximo jogador. O procedimento é repetido até que todos os jogadores tenham sido considerados e não houver alteração no vetor de ações. O resultado deste algoritmo é um equilíbrio de Nash.

Pseudocódigo 1 Algoritmo Iterativo (JACKSON, 2010).

```

1: Os jogadores escolhem a ação 1. Perfil =  $\{1,1,...,1\}$ 
2:  $Modificador = 1$ 
3: Enquanto  $Modificador = 1$  Faça                                ▷ Enquanto o Modificador for 1
4:    $Modificador = 0$ 
5:   Para  $i = 1 : N$  Faça                                          ▷ Para todos os jogadores
6:     Se  $Ações(i) = 1$  Então
7:       Crie um novo conjunto de ações onde a ação do jogador  $i = 0$ 
8:       Calcule  $M(\vec{a} : i = 1)$                                 ▷ Calcular o retorno com ação pura de  $i = 1$ 
9:       Calcule  $M(\vec{a} : i = 0)$                                 ▷ Calcular o retorno com ação pura de  $i = 0$ 
10:      Se  $M(\vec{a} : i = 0) > M(\vec{a} : i = 1)$  Então
11:         $Modificador = 1$ 
12:         $Ações(i) = 0$                                           ▷ Atualizar o vetor de ações na posição  $i$  para 0
13:      Fim Se
14:    Fim Se
15:  Fim Para
16: Fim Enquanto

```

A primeira iteração do algoritmo é ilustrada na Figura 8 para o Jogo dos Múltiplos Cruzamentos.

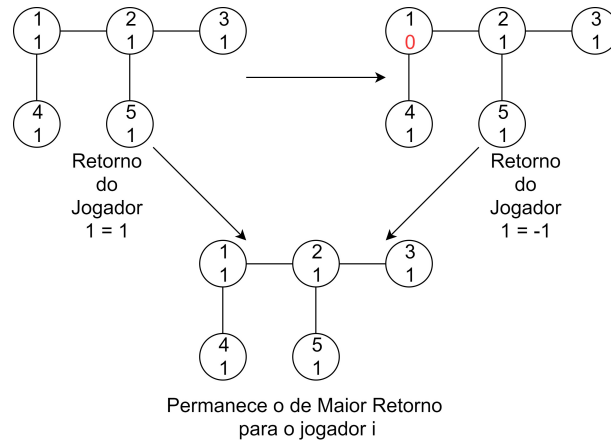


Figura 8 – Primeira iteração do Algoritmo Iterativo - Equilíbrio de Nash em grafos

O equilíbrio de Nash encontrado pelo algoritmo corresponde a todos os jogadores adotarem a ação 1. Isso ocorre após 6 ciclos do algoritmo.

O segundo algoritmo considerado foi o chamado Algoritmo TreeNash (NISAN *et al.*, 2007). Esse algoritmo assume que o grafo do jogo é uma árvore. Nesta dissertação adaptamos um dos algoritmos do TreeNash clássico: Approximate TreeNash (TreeNash Aproximado).

Tabela 7 – Exemplo de probabilidades para o algoritmo TreeNash Aproximado.

	1	2	3	4	5	6
p(0)	0	0,2	0,4	0,6	0,8	1
p(1)	1	0,8	0,6	0,4	0,2	0

O TreeNash Aproximado busca gerar uma distribuição de probabilidades para as ações puras de cada jogador. As probabilidades são discretizadas em intervalos de tamanho T . Como exemplo, a Tabela 7 apresenta seis opções ($T=0,2$) para um jogador i , sendo as linhas 1 e 2 as probabilidades do jogador adotar as ações 0 e 1, respectivamente. No exemplo dos múltiplos cruzamentos da Figura 4. Cada ação de cada jogador pode assumir os valores da Tabela 7.

Escolhendo para raiz da árvore o jogador 1, temos como primeira etapa da fase descendente do algoritmo considerar os jogadores que estão nas folhas da árvore (no exemplo, os jogadores 3, 4 e 5, nós mais distantes da raiz da árvore). Para cada folha, encontra-se seu único filho (nó diretamente ligado a ele, como na Figura 9, na cor verde). Considere pai o nó consultado e o filho sua única conexão em direção a raiz, já que o grafo é uma árvore. Para cada ação do pai e do filho é necessário calcular o retorno para o pai. Serão registradas as ações de melhores retornos.

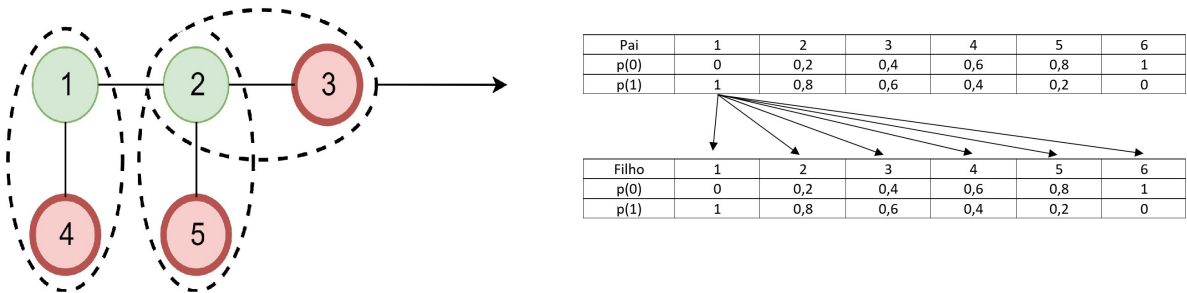


Figura 9 – Etapa 1 (descendente) do algoritmo TreeNash Aproximado.

Para a próxima etapa, os antes filhos (nó 2 no exemplo) se tornam as folhas da próxima iteração e então calcula-se os seus retornos para seus únicos filhos, como na etapa anterior. Os melhores retornos são registrados. Esta etapa é ilustrada Figura 10.

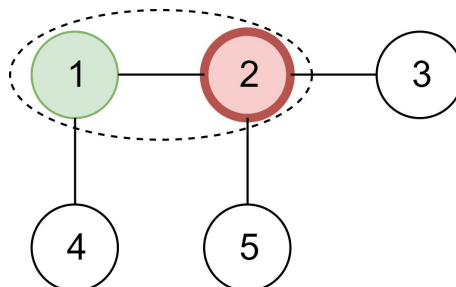


Figura 10 – Etapa 2, Final (descendente) do algoritmo TreeNash Aproximado.

Quando todos os filhos do grafo convergirem para a raiz (Figura 10), então começando pela raiz, obtém-se o melhor retorno a partir dos valores reservados na memória, isto é, para cada ação da raiz calcula-se o retorno para cada ação nas memórias de seus pais. A ação de melhor retorno será a ação da raiz, iniciando-se a fase ascendente do algoritmo. Para cada ação na memória do jogador será calculado o retorno produzido pelas memórias de seus respectivos pais. A ação de melhor retorno será a ação do jogador.

Uma modificação realizada é que, em vez de probabilidades discretas, como na Tabela 7, a determinação da melhor ação será feita resolvendo-se um Jogo Bimatriz entre pai e filho. O algoritmo TreeNash modificado é resumido no Pseudocódigo 2.

Pseudocódigo 2 Algoritmo TreeNash Modificado.

```

1: Entre com um grafo  $(G, V)$  em árvore
2: Procure as folhas da árvore:  $Folhas = (Folha_1, \dots, Folha_N)$ 
3: Enquanto  $Folhas \neq Raiz$  Faça ▷ Enquanto o vetor de folhas não for a raiz
4:    $Filhos = []$  ▷ Inicialize um vetor de filhos vazio
5:   Para  $i = 1: \text{size}(Folhas)$  Faça
6:     Ache o filho de  $Folha_i$ :  $Filho_i.Filhos = [Filhos, Filho_i]$ 
7:     Resolva um jogo bimatriz entre  $Folha_i$  e  $Filho_i$ . A solução será a melhor
       resposta de  $Folha_i$  para  $Filho_i$  e será a estratégia mista de  $Folha_i$ .
8:   Fim Para
9:    $Folhas \leftarrow Filhos$  ▷ As novas folhas são os filhos da iteração anterior
10: Fim Enquanto
11: Encontre os pais da raiz.  $PaisRaiz = (Pai_1, \dots, Pai_N)$ 
12:  $MelhorRespostaRaiz = -\infty$ 
13: Para  $i = 1: \text{size}(PaisRaiz)$  Faça
14:   Encontre a melhor resposta da folha  $i$  levando em consideração seus pais e seu
       único filho ( $Raiz$ ).
15:   Se  $Ganho(MelhorResposta_i) > MelhorRespostaRaiz$  Então
16:      $MelhorRespostaRaiz = Ganho(MelhorResposta_i)$ 
17:      $RespostaRaiz = MelhorResposta_i$ 
18:   Fim Se
19: Fim Para

```

Como exemplo, considere novamente o Jogo dos Múltiplos Cruzamentos. Usando o algoritmo TreeNash modificado, a solução encontrada é todos os jogadores adotarem a ação 1 (Seguir Horizontalmente). Isso ocorre pela estrutura do grafo e de retornos. O retorno global é 4.

A solução encontrada pelo algoritmo é um equilíbrio de Nash porque qualquer jogador que queira mudar sua ação unilateralmente será penalizado ou manterá seu retorno. Estruturas de retorno que penalizem mais os jogadores têm maior eficiência em jogos que envolvem múltiplas soluções. Como somente é considerada a ação do jogador, e não a de sua vizinhança, esse tipo de resultado é esperado.

3.2.2 Equilíbrio Correlacionado em Jogos Gráficos

O Equilíbrio Correlacionado, discutido no Capítulo 2, introduz um dispositivo de comunicação externo ao grafo, que irá definir probabilidades para cada ação conjunta possível na vizinhança N_i de um jogador i , ou seja, no equilíbrio correlacionado não se trabalha com a distribuição de probabilidades para as ações do jogador i , e sim com distribuições de probabilidades para ações conjuntas (perfis) da vizinhança (que inclui) o jogador i .

Considere o jogo da Figura 4 e os jogadores 1 e 2. O jogador 1 tem como vizinhos os jogadores (1,2,4). O equilíbrio correlacionado definirá probabilidades para todas as ações conjuntas envolvendo estes três jogadores. Já o jogador 2 tem como vizinhos (1,2,3,5), sendo as probabilidades distribuídas entre as ações conjuntas destes jogadores. Uma característica do equilíbrio correlacionado em jogos gráficos é a existência de jogadores que pertencem a diferentes vizinhanças, e para estes jogadores as probabilidades devem ser iguais. A probabilidade de uma ação de um jogador deve ser a mesma em qualquer distribuição de probabilidades de ações conjuntas, sempre que a ação for a mesma.

O algoritmo para cálculo de equilíbrios correlacionados em jogos gráficos tem como pré-requisito que o grafo seja uma árvore. Problemas em que o grafo não é uma árvore podem ser tratados fazendo-se um pré-processamento por algoritmos que os transformam em árvores geradoras mínimas, como o Algoritmo de Prim (PRIM, 1957), o Algoritmo de Dijkstra (DIJKSTRA, 1959), entre outros.

As desigualdades (3.16) representam os primeiros conjuntos de restrições para o equilíbrio correlacionado para todo jogador i e ações puras a e a' . O retorno do jogador i relativo a ação conjunta \vec{a}^i (no qual o jogador i usa a ação a) deve ser maior do que o obtido pelo jogador i quando este opta por outra ação pura a' . As ações dos outros jogadores permanecem a mesma. Para os jogos bimatrizes do Capítulo 2, estas restrições representam o cálculo da matriz C , expandida para N jogadores:

$$\sum_{\vec{a}^i: a_i^i=a} (M_i(\vec{a}^i) - M_i(\vec{a}^i[i : a'])) P_i(\vec{a}^i) \geq 0 \quad (3.16)$$

onde $P_i(\vec{a}^i)$ é a probabilidade de que a ação conjunta \vec{a}^i seja utilizada. O segundo conjunto de restrições é

$$P_i(\vec{a}^i) \geq 0; \quad \sum_{\vec{a}^i} P_i(\vec{a}^i) = 1, \quad (3.17)$$

onde a soma se refere a todos os perfis da vizinhança de i .

O terceiro conjunto de restrições é sobre as interseções das vizinhanças. Para todos os jogadores i e j e para toda ação conjunta \vec{y}^{ij} de i e j na interseção das vizinhanças,

$N_i \cap N_j$, devemos ter:

$$P_i(\vec{a}^{ij}) = \sum_{\vec{a}^i: \vec{a}^{ij} = \vec{y}^{ij}} P_i(\vec{a}^i) = \sum_{\vec{a}^j: \vec{a}^{ij} = \vec{y}^{ij}} P_j(\vec{a}^j) = P_j(\vec{a}^{ij}) \quad (3.18)$$

Voltando ao Jogo dos Múltiplos Cruzamentos, Figura 4, com as ações puras V e H denotadas por 0 e 1, por simplicidade, temos 36 possíveis ações conjuntas, sendo 8 da vizinhança do jogador 1 ($N_1 = (1, 2, 4)$), 16 na vizinhança do jogador 2 ($N_2 = (1, 2, 3, 5)$), 4 na vizinhança do jogador 3 ($N_3 = (2, 3)$), 4 na vizinhança do jogador 4 ($N_4 = (1, 4)$) e 4 na vizinhança do jogador 5 ($N_5 = (2, 5)$).

O primeiro conjunto de restrições (3.16) estabelece uma inequação para cada ação de cada jogador. Como são 5 jogadores com ações binárias, o número de inequações é igual a 10. Na forma matricial, a restrição (3.19) assumiria a estrutura:

$$A = \begin{bmatrix} A_1 & 0 & 0 & 0 & 0 \\ 0 & A_2 & 0 & 0 & 0 \\ 0 & 0 & A_3 & 0 & 0 \\ 0 & 0 & 0 & A_4 & 0 \\ 0 & 0 & 0 & 0 & A_5 \end{bmatrix} \geq 0 \quad (3.19)$$

onde os elementos A_1, A_2, \dots, A_5 são descritos na Tabela 8. Cada célula da tabela corresponde a uma ação conjunta da vizinhança.

Tabela 8 – Linhas da restrição (3.16) do Jogo dos Múltiplos Cruzamentos.

A_1	2	0	0	-2	0	0	0	0								
	0	0	0	0	-2	0	0	2								
A_2	2	0	0	-2	0	0	0	0	0	-2	-2	-4	0	0	0	0
	0	0	0	0	-2	0	0	2	0	0	0	0	0	2	2	4
A_3	0	0	-2	0												
	0	0	0	2												
A_4	2	0	0	0												
	0	-2	0	0												
A_5	2	0	0	0												
	0	-2	0	0												

O segundo conjunto de restrições é representado por 5 inequações, uma para cada jogador, representando as relações de probabilidades, que devem ser não negativas e somar

1. O segundo conjunto de restrições (3.17) pode ser colocado em forma matricial como:

$$B = \begin{bmatrix} B_1 & 0 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 & 0 \\ 0 & 0 & B_3 & 0 & 0 \\ 0 & 0 & 0 & B_4 & 0 \\ 0 & 0 & 0 & 0 & B_5 \end{bmatrix} = 1 \quad (3.20)$$

onde os elementos B_1, B_2, \dots, B_5 (linhas de B) são descritos na Tabela 9.

Tabela 9 – Linhas da restrição (3.17) do Jogo dos Múltiplos Cruzamentos.

B1	1	1	1	1	1	1	1	1									
B2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B3	1	1	1	1													
B4	1	1	1	1													
B5	1	1	1	1													

O terceiro conjunto de restrições (3.18), referente às interseções das vizinhanças, considera as restrições sobre ações conjuntas de jogadores nas interseções. Por exemplo, as vizinhanças dos jogadores 1 e 4 geram 4 restrições, pois a interseção das vizinhanças é (1,4) e cada jogador possui 2 estratégias puras. Dessa forma, o jogo terá um total de 48 restrições do tipo (3.18). Na Tabela 10 apresentam-se os coeficientes das restrições referentes aos jogadores 1 e 4. Observa-se que é atribuído o valor 1 nas posições em que a vizinhança do jogador 1 contém as estratégias (1,4) = (0,0), (0,1), (1,0) e (1,1), e -1 em que a vizinhança do jogador 4 contém as estratégias (1,4) = (0,0), (0,1), (1,0) e (1,1).

Tabela 10 – Coeficientes das restrições (3.18) relativas aos jogadores 1 e 4.

Vizinhança de 1 (1,2,4)									Vizinhança de 4(1,4)			
(1,4)	(0,0,0)	(0,0,1)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	1	0	1	0	0	0	0	0	-1	0	0	0
(0,1)	0	1	0	1	0	0	0	0	0	0	-1	0
(1,0)	0	0	0	0	1	0	1	0	0	-1	0	0
(1,1)	0	0	0	0	0	1	0	1	0	0	0	-1

Qualquer solução que respeite às restrições (3.16), (3.17) e (3.18) é um equilíbrio correlacionado. Um objetivo conveniente é maximizar o retorno global esperado, isto é, maximizar a função objetivo:

$$\sum_i \sum_{\vec{a}^i} P_i(\vec{a}^i) M_i(\vec{a}^i) \quad (3.21)$$

sujeito as restrições (3.16), (3.17 e (3.18).

A solução para o Jogo dos Múltiplos Cruzamentos é apresentada na Tabela 11. As probabilidades de estratégias conjuntas não indicadas foram todas iguais a zero.

Tabela 11 – Solução do Jogo dos Múltiplos Cruzamentos.

Jogador	Vizinhança	Estratégia	Probabilidade
1	(1,2,4)	(0,0,0)	0,5675
		(1,1,1)	0,4325
2	(1,2,3,5)	(0,0,0,0)	0,5675
		(1,1,1,1)	0,4325
3	(2,3)	(0,0)	0,5675
		(1,1)	0,4325
4	(1,4)	(0,0)	0,5675
		(1,1)	0,4325
5	(2,5)	(0,0)	0,5675
		(1,1)	0,4325

Comparando os retornos, o equilíbrio de Nash fornece retorno global 4, o mesmo retorno obtido por meio do equilíbrio correlacionado. Uma vantagem do equilíbrio correlacionado é fornecer retornos para todos os jogadores maiores ou iguais aos que são obtidos por meio do equilíbrio de Nash (PETERS, 2015).

3.2.3 Solução de Jogos Gráficos por Algoritmo Genético

Um dos métodos possíveis para a obtenção de equilíbrios em jogos gráficos baseia-se em Computação Evolutiva. A estrutura do algoritmo pode ser vista na Figura 11. A computação evolutiva é um ramo de pesquisa em Inteligência Artificial que propõe paradigmas para solução de problemas inspirados na Seleção Natural.

Um dos algoritmos representativos da computação evolutiva é o algoritmo genético (HOLLAND, 1975). Este algoritmo baseia-se na evolução de indivíduos por meios computacionais como forma de resolver problemas. O algoritmo genético clássico foi concebido com codificação binária, podendo ser adaptado para diversos tipos de representação mantendo sua estrutura clássica.

Uma das entradas do algoritmo é o número de indivíduos da população, que especifica o tamanho do universo de possibilidades que serão evoluídas. Cada indivíduo representa uma possível solução para o problema, sendo necessário escolher uma forma de codificação do indivíduo, que pode ser por vetores binários, reais, máquinas de estados ou qualquer representação que retorne uma medida (*fitness*) de quão boa é a resposta produzida pelo indivíduo a partir de uma função, conhecida como função de fitness.

O critério de parada é um aspecto importante a ser considerado, pois o algoritmo genético não garante otimalidade global quando aplicado a problemas de otimização. O critério de parada pode ser o número de gerações, isto é, o número de vezes que a população evoluirá, ou a diferença entre o fitness do melhor indivíduo da população anterior e da atual, entre outras.

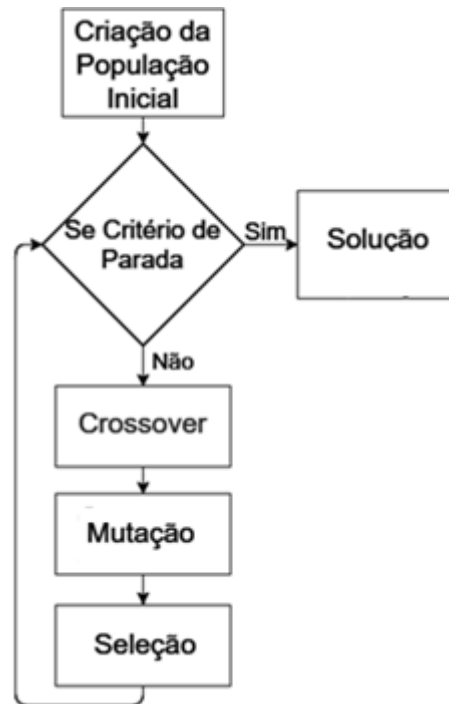


Figura 11 – Estrutura de um Algoritmo Genético.

O crossover é um processo que envolve dois ou mais indivíduos da população, chamados de pais, aos quais se dá chance de gerar filhos, que mantêm características dos pais. O objetivo é achar grupos de pais que têm boas características e então gerar novos indivíduos que as mantenham.

A mutação é um processo local, centrado no indivíduo, no qual cada parte do indivíduo tem a possibilidade de sofrer pequenas mudanças, que podem levar o indivíduo a produzir um melhor retorno (fitness).

No processo de seleção, escolhem-se os indivíduos com os melhores retornos para a próxima população. Essa escolha pode ser gulosa, escolhendo-se sempre os melhores indivíduos, ou pode manter indivíduos piores, porém com possibilidade de melhoria nas próximas gerações.

O Algoritmo Genético que é desenvolvido nesta dissertação tem como objetivo encontrar soluções com maiores retornos globais, sem preocupação com a rigidez imposta pelos equilíbrios de Nash e Correlacionado. A motivação para essa escolha é a procura de soluções que de modo geral produzem bons retornos, mesmo que um jogador seja penalizado em um certo momento.

O algoritmo genético é heurístico, e portanto não garante convergência para uma solução ótima global. Porém fornece, com certo nível de inteligência, respostas de maneira rápida e de fácil aplicação.

Voltando ao Jogo dos Múltiplos Cruzamentos da Figura 4, temos como forma de

representação um vetor com 5 posições binárias, em que cada posição (locus) i representa a ação do jogador i (Tabela 12). Foram iniciados 'Tamanho da População' indivíduos, representando diferentes ações conjuntas, de forma aleatória, sem nenhum conhecimento prévio da possível solução. O tamanho da população é indicado na Tabela 13.

Tabela 12 – Estrutura de representação do indivíduo - Jogo dos Múltiplos Cruzamentos.

Ação Jogador 1	Ação Jogador 2	Ação Jogador 3	Ação Jogador 4	Ação Jogador 5
----------------	----------------	----------------	----------------	----------------

O critério de parada para o algoritmo foi o número de gerações evoluídas pelo algoritmo. O número de gerações utilizado também está indicado na Tabela 13.

O método de *crossover* utilizado para o problema é o Método de N Pontos, pelo qual cada dupla de pais produz dois filhos idênticos aos pais, porém em N posições, de quantidade e posições aleatórias, o filho 1 recebe as informações do pai 2 e o filho 2 a informação do pai 1. É escolhido o tamanho da população de pais em potencial, de forma aleatória, sendo que cada par de pais tem uma probabilidade de 'Porcentagem de Crossover' de gerar filhos (Tabela 13).

O método de mutação utilizado é a Mutação de 1 Ponto. Esse método consiste em atribuir a cada posição de cada jogador uma probabilidade 'Porcentagem de Mutação' de mudar seu valor. (Tabela 13).

O método de seleção utilizado é o torneio com salvacionismo. O método consiste em selecionar da população total (pais, filhos e mutantes) o número de indivíduos especificado em 'Tamanho da População'. Para cada escolha é sorteado 'Número Indivíduos Torneio' indivíduos de forma aleatória para um torneio, no qual o melhor indivíduo, com maior retorno, é o vencedor e permanece na próxima geração. Tanto o vencedor quanto os perdedores voltam para a população para outros torneios. A relação entre o número de indivíduos por torneio e o tamanho da população controla a pressão seletiva do processo; se a relação for igual a 1, somente o melhor indivíduo vai sobreviver. Como os indivíduos do torneio são escolhidos aleatoriamente, é possível que o melhor indivíduo não seja escolhido em nenhum torneio, e portanto não vá para a próxima geração. Para solucionar esse problema é utilizado o salvacionismo, que é a adição do melhor indivíduo da população anterior na próxima população.

Tabela 13 – Dados utilizados na simulação - Jogo dos Múltiplos Cruzamentos.

	Valor
Tamanho da População	25
Gerações	100
Porcentagem de Crossover	50%
Porcentagem de Mutação	10%
Número de Indivíduos por Torneio	3

Para o cálculo do retorno foi utilizada a soma dos retornos de todos os jogadores (Tabela 6). Para o problema com a estrutura da Figura 4 e os valores da Tabela 13 foram encontrados dois possíveis equilíbrios: todos os jogadores adotando a ação 1 (mesma solução do equilíbrio de Nash com retorno 4) e todos os jogadores adotando a ação 0 (estratégia até agora não encontrada, sendo também um equilíbrio de Nash, com o retorno global de 4). Um destaque é a rapidez com que encontra-se esta solução. Na Figura 12 ilustra-se a evolução do algoritmo, que foi implementado em MATLAB v2009. Pode-se observar que na terceira geração já há indivíduos com retorno máximo; na décima primeira o retorno médio da população já está próximo ao máximo, o que indica convergência de todas as soluções para a de retorno máximo. Em exemplos mais complexos e dinâmicos a escolha de um 'Número de Indivíduos por Torneio' adequado pode acelerar ou retardar a convergência, mantendo a diversidade da população.

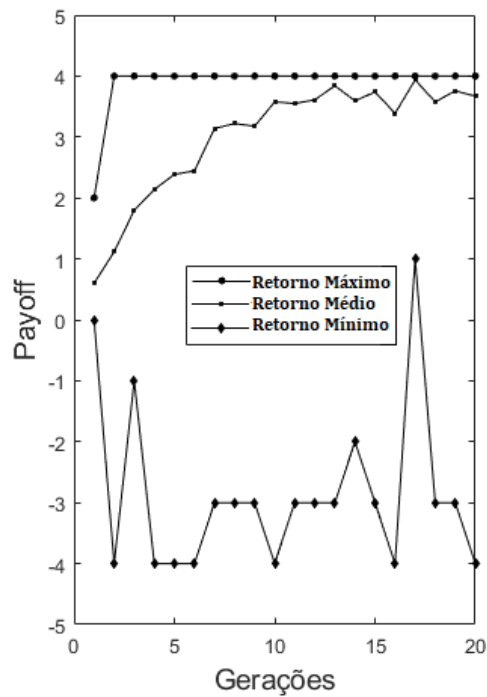


Figura 12 – Evolução do Algoritmo Genético - Jogo dos Múltiplos Cruzamentos.

4 Controle de Sistemas Dinâmicos Multiagentes

O objetivo central da dissertação é o controle de sistemas dinâmicos multiagentes em grafos usando a Teoria dos Jogos e, mais especificamente, o equilíbrio correlacionado. No caso de sistemas dinâmicos com dois ou mais agentes, busca-se coordenar os objetivos individuais e do grupo de agentes por meio de protocolos (regras) de controle distribuídos, isto é, implementados a partir de informações locais, fornecidas pela vizinhança de cada um dos agentes.

4.1 Algoritmos de Controle em Grafo

Em jogos gráficos, a topologia do grafo de comunicação é de vital importância. Desejamos alcançar uma sincronização no sentido do objetivo global ao mesmo tempo que otimizamos especificações de desempenho locais nos agentes. Para tanto, utilizaremos conceitos e métodos de Teoria dos Jogos, em particular, o conceito de equilíbrio Correlacionado.

Em (LEWIS *et al.*, 2013) o problema de controle de sistema multiagentes é tratado de forma abrangente. Nesta referência são desenvolvidos métodos para sistemas dinâmicos gerais baseados na resolução de equações de Riccati. Tanto sistemas contínuos quanto sistemas discretizados são considerados. Resultados para sistemas de primeira, segunda e ordens superiores são apresentados.

Jogos Diferenciais são jogos em que o comportamento de cada jogador é (tipicamente) descrito por uma equação diferencial (FLEMING, 1957), (FRIEDMAN, 1970). A ação de cada jogador é uma função do tempo, por meio da qual o jogador procura otimizar o seu funcional. Equações diferenciais e funcionais dependem do conjunto das ações de todos os jogadores.

Nesta dissertação, consideramos agentes (jogadores) descritos por dinâmicas lineares e invariantes no tempo, na forma geral:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{4.1}$$

sendo $x(t) \in \mathbb{R}^n$ o vetor de estados, $u(t) \in \mathbb{R}^m$ o vetor de entradas e $y(t) \in \mathbb{R}^p$ o vetor de saídas do agente; A , B , C e D são matrizes de dimensões apropriadas (BONATTI *et al.*, 2016). Um subíndice i pode ser acrescentado a variáveis e matrizes para especificar o

agente i . O estado do agente i então seria $x_i(t) \in \mathbb{R}^{n_i}$ e assim por diante. Nas simulações realizadas, consideramos somente sistemas SISO (Single Input - Single Output), isto é, $m = p = 1$. Para implementação de tarefas de simulação envolvendo sistemas dinâmicos utilizamos os softwares MATLAB/Simulink v2009.

Nos capítulos anteriores desenvolvemos algoritmos para encontrar equilíbrios de Nash e Correlacionado em jogos estáticos. Neste capítulo abordaremos o problema de controle dinâmico por meio de um algoritmo discreto sobre uma estrutura de grafo. Inicialmente busca-se entender os conceitos de equilíbrio vistos anteriormente no sentido de também contemplar jogos dinâmicos.

O algoritmo proposto baseia-se em medidas de variação dos estados dos jogadores causadas por suas ações, para cálculo da matriz de retornos e solução de um jogo a cada instante de tempo, como na Figura 13.

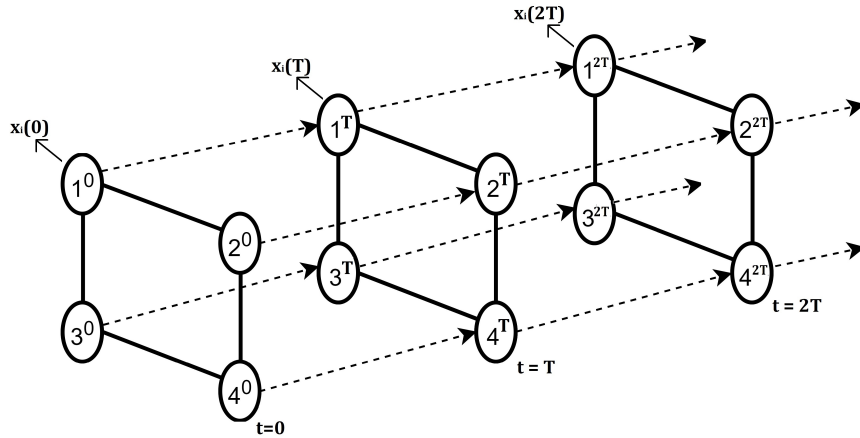


Figura 13 – Modelo do Algoritmo de Controle - Variação dos Estados.

O algoritmo, descrito no Pseudocódigo 3 e diagrama da Figura 14, consiste na realização de jogos estáticos e atualização dos estados em instantes discretos de tempo. Inicialmente calculam-se as matrizes de retorno $M_i(\vec{a})$ de cada jogador i utilizando a ação \vec{a} prevista, levando em consideração o estado atual. A partir das matrizes $M_i(\vec{a})$ resolve-se um jogo estático no tempo atual. A escolha da ação tomada pelo jogador é determinada pelo Algoritmo da Roleta (LIPOWSKI; LIPOWSKA, 2012), um algoritmo para seleção de ações proporcionais às suas probabilidades. O Algoritmo de Roleta é apresentado no Pseudocódigo 4. As ações determinadas pelo Algoritmo de Roleta são implementadas sempre que o erro entre os estados dos jogadores e o valor de consenso a atingir for maior que $\epsilon > 0$ especificado.

O Equilíbrio de Nash tem como premissa a determinação da melhor ação de cada jogador tendo em vista as melhores ações dos jogadores em sua vizinhança. Esse comportamento é compatível com objetivos individuais de cada jogador, mas não com objetivos globais. Já o Equilíbrio Correlacionado tem como característica a existência de um agente

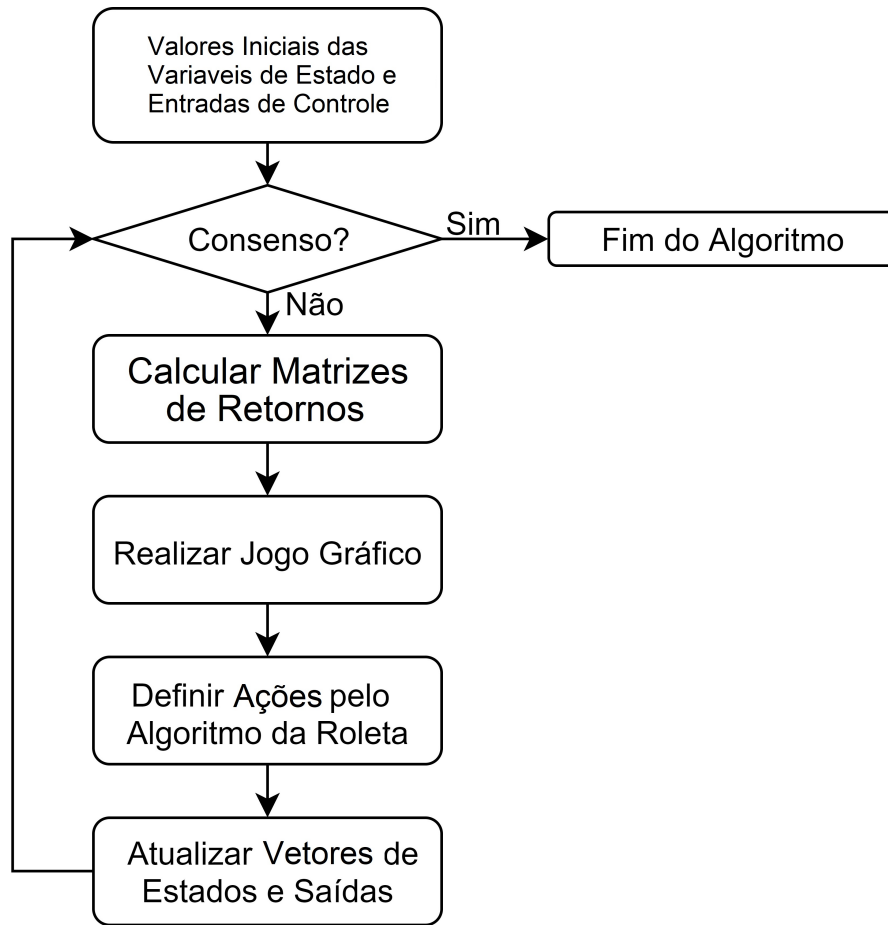


Figura 14 – Diagrama do Algoritmo.

Pseudocódigo 3 Algoritmo de Controle de Sistemas Multiagentes.

- 1: Entre com os valores dos estados e entradas iniciais.
- 2: Cálculo do erro inicial
- 3: **Enquanto** $Erro > \epsilon$ **Faça**
- 4: Calcule a matriz M ▷ Calcule as matrizes de retornos dos jogadores
- 5: Realize o jogo estático cooperativo com a matriz M .
- 6: Determine ações através do Algoritmo de Roleta
- 7: Atualize os valores das variáveis do sistema em função das ações determinadas anteriormente
- 8: Calcule o novo erro.
- 9: **Fim Enquanto**

mediador que recomenda ações conjuntas para toda uma vizinhança. Como as vizinhanças acabam tendo interseções, esse agente mediador utiliza informações sobre todos os jogadores, ainda que indiretamente, o que é vantajoso em termos de objetivos globais, não somente individuais.

A seguir aplicaremos o algoritmo de controle multiagentes a um problema dinâmico, que consiste em evoluir os estados dos jogadores para estados pré-determinados. Detalhes sobre como calcular as matrizes $M_i(\vec{a})$ e atualizar os estados dos jogadores são

Pseudocódigo 4 Algoritmo da Roleta de um Cassino.

-
- 1: Entre com o conjunto de probabilidades W e o número de ações N
 - 2: $W_m = \max(W)$ ▷ Valor máximo das probabilidades
 - 3: $index = \text{round}(\text{rand} * N)$ ▷ Escolha índice de maneira aleatória
 - 4: $F = 10 * W_m * \text{rand}$ ▷ Força aleatória do Crupiê aplicada na roleta
 - 5: **Enquanto** $F > W(index)$ **Faça** ▷ Enquanto a força mantiver a roleta girando
 - 6: $F = F - W(index)$ ▷ Redução da força por atrito até que a roleta pare
 - 7: $index = (index + 1) \bmod(N)$ ▷ Passa para o próximo slot da roleta, sendo que no último índice volte para o índice 1
 - 8: **Fim Enquanto**
 - 9: ProbabilidadeEscolhida = índice
-

apresentados.

4.1.1 Estados Regulados

Como exemplo para aplicação do algoritmo considere um grafo com 5 jogadores, como ilustrado na Figura 15. Cada jogador apresenta a mesma dinâmica:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u \quad (4.2)$$

Assume-se que cada jogador tem acesso aos seus estados e aos estados dos agentes vizinhos.

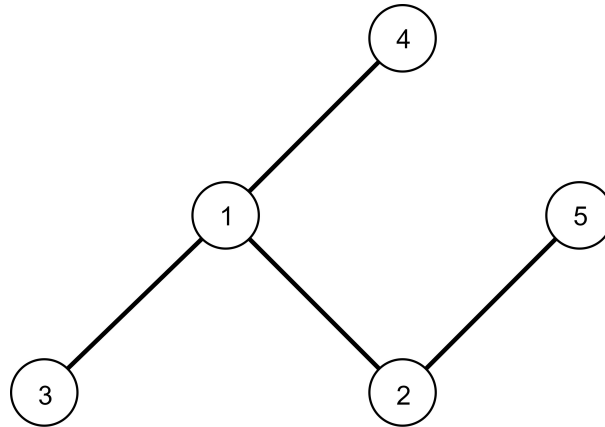


Figura 15 – Exemplo de controle dinâmico.

O algoritmo é alimentado com os valores iniciais das variáveis de estados (x_{1i}^0 e x_{2i}^0) e da entrada de controle (u_i^0) de cada sistema i . Consideramos duas ações para cada jogador: aumentar ou diminuir a entrada de controle u_i .

Dessa forma, dado o objetivo de levar todos os jogadores para um mesmo estado,

calcula-se a distância dos estados atuais do jogador i para os de seus vizinhos:

$$\Delta_i = \sum_{s=1}^2 \sum_{j \in N_i} |x_{si} - x_{sj}| \quad (4.3)$$

Para cada jogador e para cada ação conjunta \vec{a} possível (no exemplo são $2^5 = 32$ ações possíveis) determina-se os estados futuros do sistema para as ações conjuntas possíveis, gerando os estados $x_{1i}^{\vec{a}}$ e $x_{2i}^{\vec{a}}$. Por exemplo, o jogador 2 tem como vizinhos (1,2,5). Com a ação (0,1,1) em que o jogador 1 diminui u_1 , o jogador 2 aumenta u_2 e o jogador 5 aumenta u_5 , o jogador 2 terá como estado $x_{12}^{(0,1,1)}$ e $x_{22}^{(0,1,1)}$. Para cada um desses estados é calculada uma nova diferença:

$$\Delta_i^{\vec{a}} = \sum_{s=1}^2 \sum_{j \in N_i} |x_{si}^{\vec{a}} - x_{sj}^{\vec{a}}| \quad (4.4)$$

Como o objetivo é diminuir a distância entre os estados, ações que produzem aumento da distância levam a retornos ruins, enquanto que ações que diminuem a diferença levam a retornos bons. Dessa forma, definimos o retorno da ação \vec{a} para o jogador i é como:

$$M_i(\vec{a}) = \Delta_i - \Delta_i^{\vec{a}} \quad (4.5)$$

Se $\Delta_i^{\vec{a}} > \Delta_i$, a ação aumenta a distância e tem retorno negativo proporcional ao aumento da diferença. Caso contrário, a ação terá retorno proporcional à diminuição da diferença produzida.

A partir dos cálculos dos retornos, realiza-se um jogo gráfico estático como no capítulo anterior. Esse jogo retorna um equilíbrio correlacionado, que são probabilidades para ações conjuntas. Utiliza-se o algoritmo da Roleta para a escolha das ações puras.

Com a ação escolhida, cada jogador atualiza suas variáveis a partir do aumento ou diminuição da entrada de controle. Essa atualização é feita proporcionalmente à distância entre os jogadores. Quanto mais longe os jogadores estiverem, maior o incremento na entrada. Um refinamento local é realizado quando a diferença é pequena. O sinal é negativo se a ação escolhida for zero, e positiva caso contrário:

$$u_i^{t+1} = u_i^t \pm G(\Delta_i - \Delta_i^{\vec{a}}) \quad (4.6)$$

O ganho G é um ganho proporcional que ajusta a taxa de atualização da entrada de controle. Valores altos tendem a aumentar a taxa de convergência, porém podem gerar oscilações no sistema. Nas simulações a seguir, utilizamos $G = 0, 2$.

Após a atualização dos estados, retorna-se ao cálculo da matrizes de retornos, caso as saídas não tenham atingido o consenso. Caso o consenso tenha sido encontrado, o algoritmo é finalizado.

Para o exemplo com os sistemas tendo dinâmica definida pela equação (4.2) e estrutura da Figura 15, obtemos o resultado da Figura 16. O intervalo de amostragem foi de 0,1 segundos. O critério de consenso foi que a distancia entre todos os estados de todos os jogadores fosse menor que 0,1. Pode-se observar que o consenso foi encontrado por volta de 3 segundos, embora os estados de todos os agentes tenham se aproximado a partir de 1 segundo para estados iniciais em $[2, 1, 0, -1, -2]$ e $[2, 1, 0, -1, -2]$, respectivamente.

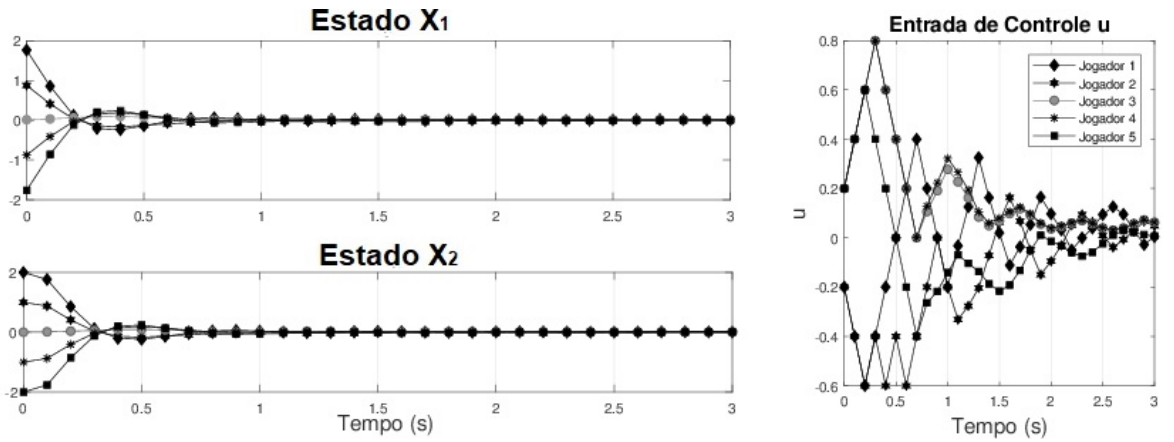


Figura 16 – Resultados do algoritmo de controle multiagentes.

Para 50 valores aleatórios de entradas entre -2 e 2 o tempo médio para obtenção do consenso foi de 3,3 segundos, como ilustrado na Figura 17.

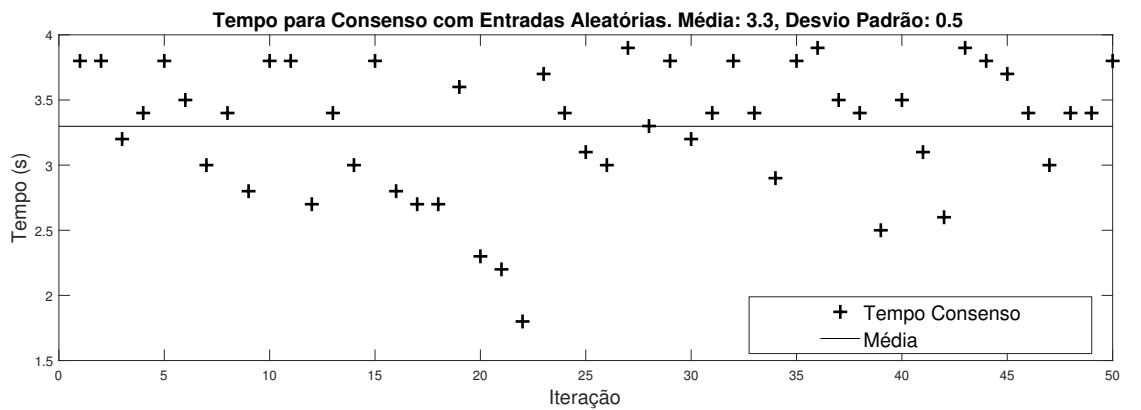


Figura 17 – Tempo dos ensaios para 50 entradas diferentes e aleatórias.

4.1.2 Consenso de Estados em Relação a um Líder

Na análise anterior, busca-se reduzir a distancia entre os estados dos agentes para zero, sendo os estados de convergência dependentes somente da estrutura do grafo, do valor inicial das variáveis de estado e de entrada de controle. Apesar deste ser um objetivo válido

(LEWIS *et al.*, 2013) (NISAN *et al.*, 2007), uma especificação dos estados de equilíbrio pode refletir melhor os objetivos globais do sistema multiagente.

A introdução de um nó líder, também chamado de nó zero, que pode ser um dos próprios nós do grafo, cumpre esta finalidade. O nó líder pode estar conectado a todos os jogadores ou somente a um subconjunto de jogadores, como ilustrado na Figura 18.

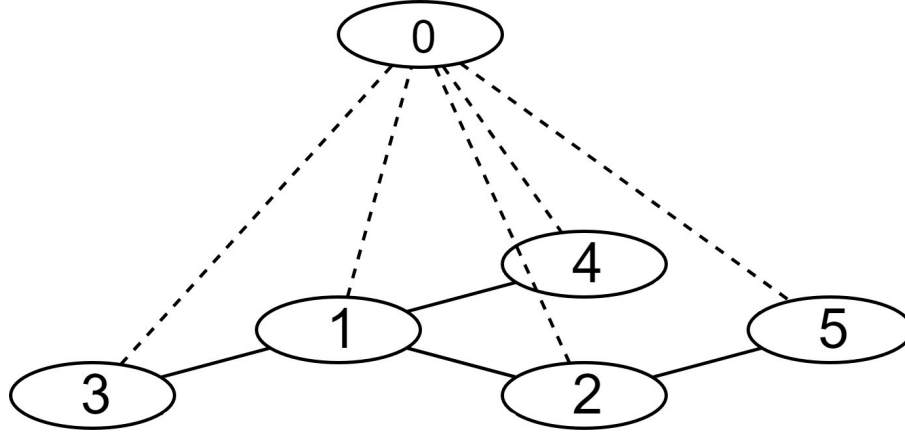


Figura 18 – Representação da introdução de um nó líder.

A introdução no nó líder modifica a estrutura do grafo e consequentemente o fluxo de informações no grafo. Ao se conectar mais nós ao líder, mais rápida é o impacto da informação do líder sobre a dinâmica dos nós, gerando um aumento na velocidade de sincronização dos estados dos agentes perante a missão proposta pelo líder.

Nos nós pertencentes à vizinhança do líder ($i \in N_0$), os cálculos das distâncias atual (4.7) e para a ação conjunta (4.8) tem a adição de mais um termo:

$$\Delta_i = \sum_{s=1}^2 \sum_{j \in N_i} G_V |x_{si} - x_{sj}| + \sum_{s=1}^2 G_L |x_{si} - x_{s0}| = \Delta_{iV} + \Delta_{iL} \quad (4.7)$$

$$\Delta_i^{\vec{a}} = \sum_{s=1}^2 \sum_{j \in N_i} G_V |\vec{x}_{si} - \vec{x}_{sj}| + \sum_{s=1}^2 G_L |\vec{x}_{si} - \vec{x}_{s0}| = \Delta_{iV}^{\vec{a}} + \Delta_{iL}^{\vec{a}} \quad (4.8)$$

Os ganhos G_V e G_L são ganhos proporcionais para os objetivos de reduzir a distância entre os agentes e ao líder, respectivamente. Quanto maior a relação G_L/G_V , mais os agentes na vizinhança do líder escolherão ações que os aproximem da trajetória do líder.

Utilizando o mesmo algoritmo do Pseudocódigo 3, com os mesmos sistemas descritos em (4.2) e mesma estrutura da Figura 15, e assumindo que apenas o nó 3 está conectado ao líder, que comanda o seguimento de estados iguais a 1, obtêm-se os resultados da Figura 19. Consideramos $G_L/G_V = 1$. O seguimento de referências tende a ser

mais rápido na medida que mais nós estejam conectados ao líder. No caso mais geral, a dinâmica do líder serve de referencia para os demais jogadores.

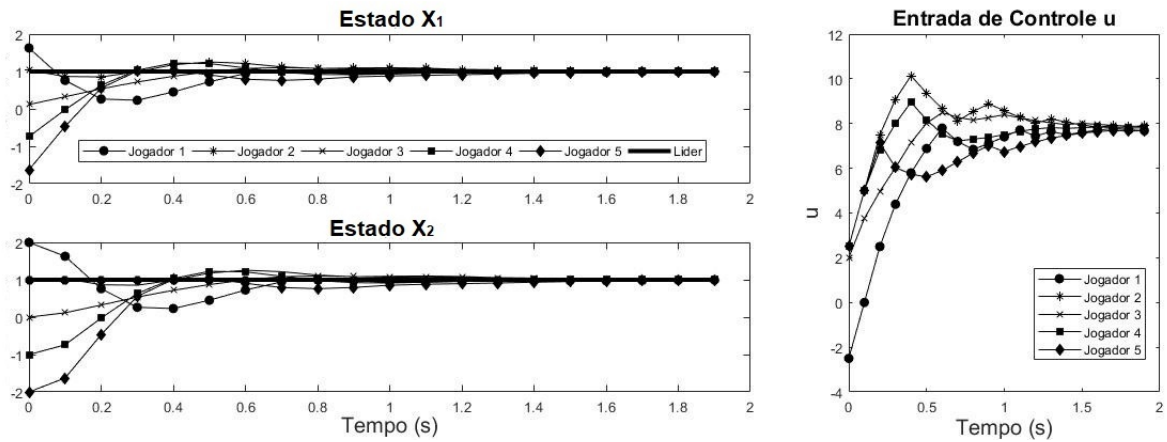


Figura 19 – Resultado da introdução de um nó líder.

Para 50 valores aleatórios de entradas entre -2 e 2 para teste do algoritmo, obtemos média de 4,53 segundos para consenso, como ilustrado na Figura 20.

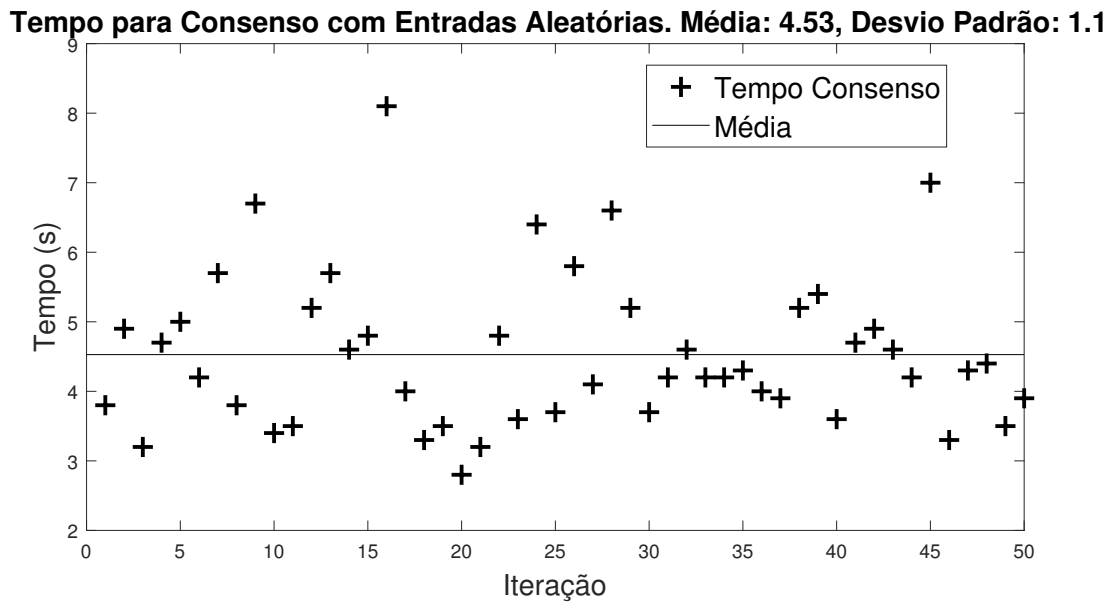


Figura 20 – Resultado para 50 entradas aleatórias para a presença do nó líder.

Os estados do líder não precisam ser constantes. Para o mesmo exemplo anterior com estados variando como uma senoide de frequência de $1/5$ radianos, obtivemos os resultados da Figura 21. Para frequências maiores, existe a necessidade de adequar o ganho da Equação 4.6 para que a taxa de atualização dos estados se adeque a variação da missão.

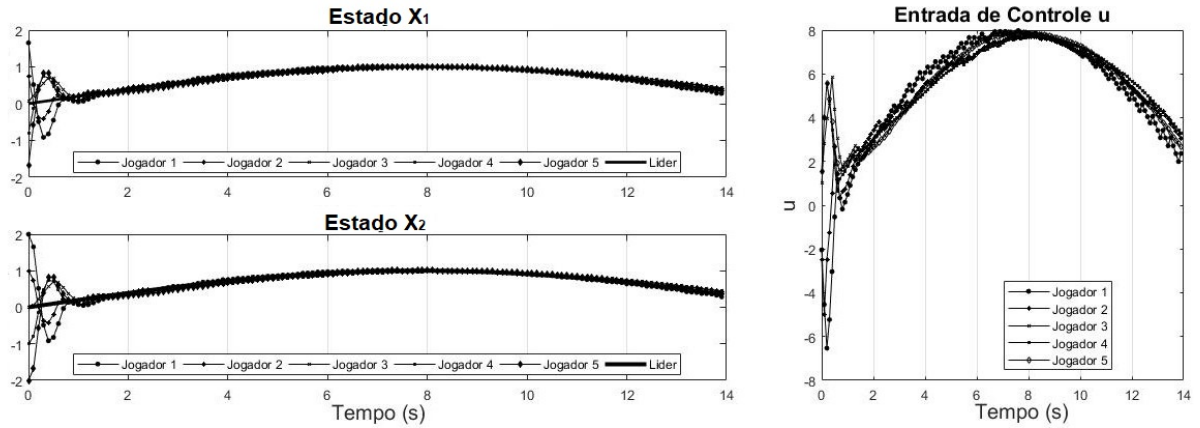


Figura 21 – Resultado da introdução de um nó líder e trajetória variante.

4.1.3 Estados Ruidosos

Para testar o funcionamento do algoritmo na presença de ruído, no exemplo inicial adicionou-se uma variável aleatória de média zero e desvio padrão 0,01 a cada saída de cada um dos agentes. Os resultados são apresentados na Figura 22.

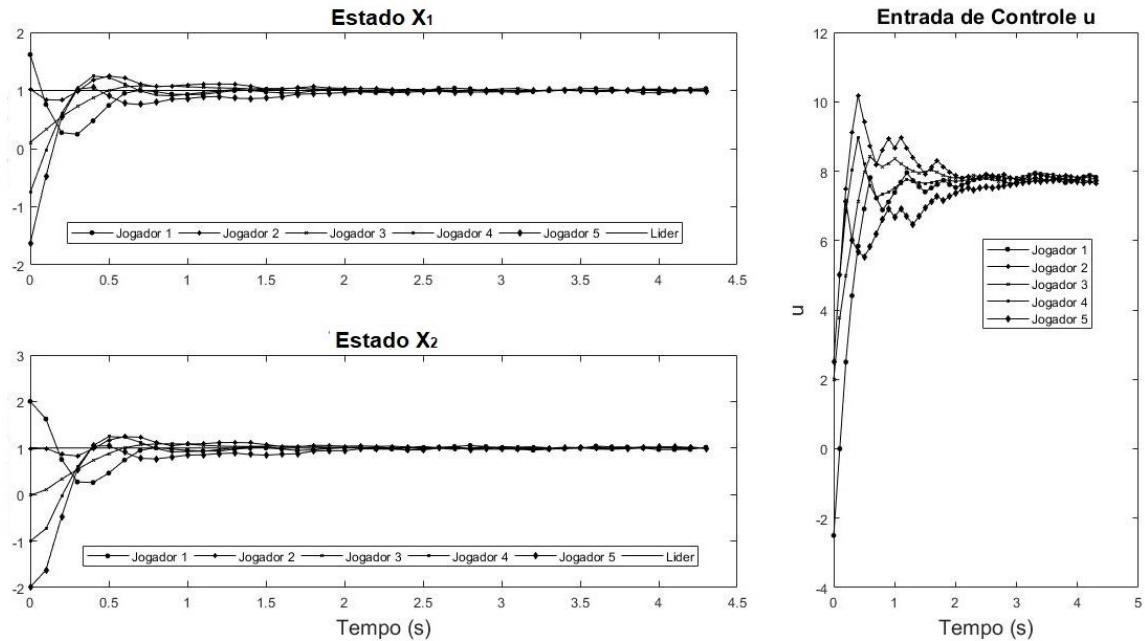


Figura 22 – Resultado da introdução de ruído e valores constantes.

O algoritmo não convergiu segundo os critérios de consenso em virtude do ruído, porém convergiu para valores muito próximos dos valores de referência, mostrando que o algoritmo consegue lidar com a presença de ruídos.

Além de uma missão pré-definida o nó líder pode ser representado por um sistema com dinâmica própria, a ser seguida pelos demais. Quanto mais distinta for a dinâmica do líder em relação às dos demais, maior dificuldade existirá para obtenção do consenso.

5 Conclusão

Nesta dissertação abordamos o problema de controle de sistemas dinâmicos multiagentes conectados por redes de comunicação. Na abordagem do problema utilizamos a Teoria dos Jogos, mais especificamente Jogos Gráficos.

No Capítulo 2 apresentamos elementos básicos sobre Teoria dos Jogos, com ênfase nos conceitos de Equilíbrio de Nash e Equilíbrio Correlacionado.

No Capítulo 3 revimos os Equilíbrios de Nash e Correlacionado à luz de Jogos Gráficos, nos quais grafos estabelecem a estrutura de comunicação entre os jogadores. Para tanto, fez-se necessário um estudo prévio sobre elementos da Teoria Algébrica de Grafos.

Algoritmos especializados para obtenção de equilíbrios de Nash e Correlacionado foram implementados e testados. De uma maneira geral, algoritmos que produzem equilíbrios correlacionados tendem a refletir melhor a natureza de interação existem em Jogos Gráficos, e em sistemas multiagentes, em particular. A existência de um 'mediador' das ações conjuntas de jogadores em vizinhanças do grafo parece ir ao encontro da busca por atender objetivos locais e globais, presentes na maioria das formulações que envolvem sistemas de controle multiagentes.

No Capítulo 4 aplicamos os resultados obtidos nos capítulos anteriores ao problema de sincronizar o comportamento de sistemas dinâmicos conectados por redes de comunicação. A literatura sobre controle de sistemas dinâmicos multiagentes é toda ela centrada na noção de Equilíbrio de Nash. Nesta dissertação adotamos a noção de Equilíbrio Correlacionado, obtendo sucesso na resolução do problema de assegurar consenso entre os agentes, ou dos agentes em relação a um agente líder. Resultados computacionais demonstraram a viabilidade da noção de Equilíbrio Correlacionado, em substituição à de Equilíbrio de Nash, em Jogos Gráficos envolvendo múltiplos agentes dinâmicos.

Como proposta para trabalhos futuros, uma melhor integração da noção de Equilíbrio Correlacionado com a Teoria de Controle Ótimo (em especial, a partir da introdução de critérios de desempenho locais para os jogadores) visando a obtenção de leis de controle em malha fechada, e uma análise sobre viabilidade computacional dos algoritmos desenvolvidos em condições mais realistas (práticas), são os temas que consideramos mais promissores.

Referências

- AUMANN, R. J. Correlated equilibrium as an expression of bayesian rationality. *Econometrica: Journal of the Econometric Society*, JSTOR, p. 1–18, 1987. Citado na página 18.
- BONATTI, I. S.; LOPES, A.; PERES, P. L. D.; AGULHARI, C. M. *Linearidade em sinais e sistemas*. [S.l.]: Editora Blucher, 2016. Citado na página 40.
- CHEN, J.; CAO, X.; CHENG, P.; XIAO, Y.; SUN, Y. Distributed collaborative control for industrial automation with wireless sensor and actuator networks. *IEEE Transactions on Industrial Electronics*, IEEE, v. 57, n. 12, p. 4219–4230, 2010. Citado na página 12.
- CRUZ, D.; MCCLINTOCK, J.; PERTEET, B.; ORQUEDA, O. A.; CAO, Y.; FIERRO, R. Decentralized cooperative control-a multivehicle platform for research in networked embedded systems. *IEEE control systems*, IEEE, v. 27, n. 3, p. 58–78, 2007. Citado na página 12.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische mathematik*, Springer, v. 1, n. 1, p. 269–271, 1959. Citado na página 33.
- FELE, F.; MAESTRE, J. M.; CAMACHO, E. F. Coalitional control: Cooperative game theory and control. *IEEE Control Systems*, IEEE, v. 37, n. 1, p. 53–69, 2017. Citado na página 13.
- FLEMING, W. A note on differential games of prescribed duration. *Contributions to the Theory Games*, p. 407–412, 1957. Citado na página 40.
- FRIEDMAN, A. Existence of value and of saddle points for differential games of pursuit and evasion. *Journal of Differential Equations*, Elsevier, v. 7, n. 1, p. 92–110, 1970. Citado na página 40.
- HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: U Michigan Press, 1975. Citado na página 36.
- JACKSON, M. O. *Social and economic networks*. [S.l.]: Princeton University Press, 2010. Citado 2 vezes nas páginas 29 e 30.
- LEWIS, F. L.; ZHANG, H.; HENGSTER-MOVRIC, K.; DAS, A. *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. [S.l.]: Springer Science & Business Media, 2013. Citado 3 vezes nas páginas 12, 40 e 46.
- LIPOWSKI, A.; LIPOWSKA, D. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 391, n. 6, p. 2193–2196, 2012. Citado na página 41.
- LUCE, R. D.; RAIFFA, H. *Games and decisions: Introduction and critical survey*. [S.l.]: Courier Corporation, 2012. Citado na página 13.

- MURRAY, R. M. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, American Society of Mechanical Engineers, v. 129, n. 5, p. 571–583, 2007. Citado na página 12.
- NASH, J. Non-cooperative games. *Annals of mathematics*, JSTOR, p. 286–295, 1951. Citado na página 16.
- NISAN, N.; ROUGHGARDEN, T.; TARDOS, E.; VAZIRANI, V. V. *Algorithmic game theory*. [S.l.]: Cambridge University Press Cambridge, 2007. v. 1. Citado 2 vezes nas páginas 30 e 46.
- PETERS, H. *Game theory: A Multi-leveled approach*. [S.l.]: Springer, 2015. Citado 4 vezes nas páginas 15, 16, 20 e 36.
- PRIM, R. C. Shortest connection networks and some generalizations. *Bell Labs Technical Journal*, Wiley Online Library, v. 36, n. 6, p. 1389–1401, 1957. Citado na página 33.
- QUIJANO, N.; OCAMPO-MARTINEZ, C.; BARREIRO-GOMEZ, J.; OBANDO, G.; PANTOJA, A.; MOJICA-NAVA, E. The role of population games and evolutionary dynamics in distributed control systems: The advantages of evolutionary game theory. *IEEE Control Systems*, IEEE, v. 37, n. 1, p. 70–97, 2017. Citado na página 13.
- REYNOLDS, C. W. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH computer graphics*, ACM, v. 21, n. 4, p. 25–34, 1987. Citado na página 12.
- VAMVOUDAKIS, K. G.; MODARES, H.; KIUMARSI, B.; LEWIS, F. L. Game theory-based control system algorithms with real-time reinforcement learning: how to solve multiplayer games online. *IEEE Control Systems*, IEEE, v. 37, n. 1, p. 33–52, 2017. Citado na página 13.
- WANG, X.; YADAV, V.; BALAKRISHNAN, S. Cooperative uav formation flying with obstacle/collision avoidance. *IEEE Transactions on control systems technology*, IEEE, v. 15, n. 4, p. 672–679, 2007. Citado na página 12.